

Article

Stochastic Recursive Gradient Support Pursuit and Its Sparse Representation Applications

Fanhua Shang ^{*}, Bingkun Wei, Yuanyuan Liu, Hongying Liu , Shuang Wang and Licheng Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China; bkwei028@gmail.com (B.W.); yylliu@xidian.edu.cn (Y.L.); hylu@xidian.edu.cn (H.L.); shwang@mail.xidian.edu.cn (S.W.); lchjiao@mail.xidian.edu.cn (L.J.)

* Corresponding author: fhshang@xidian.edu.cn

Received: 7 July 2020; Accepted: 27 August 2020; Published: 30 August 2020



Abstract: In recent years, a series of matching pursuit and hard thresholding algorithms have been proposed to solve the sparse representation problem with ℓ_0 -norm constraint. In addition, some stochastic hard thresholding methods were also proposed, such as stochastic gradient hard thresholding (SG-HT) and stochastic variance reduced gradient hard thresholding (SVRGHT). However, each iteration of all the algorithms requires one hard thresholding operation, which leads to a high per-iteration complexity and slow convergence, especially for high-dimensional problems. To address this issue, we propose a new stochastic recursive gradient support pursuit (SRGSP) algorithm, in which only one hard thresholding operation is required in each outer-iteration. Thus, SRGSP has a significantly lower computational complexity than existing methods such as SG-HT and SVRGHT. Moreover, we also provide the convergence analysis of SRGSP, which shows that SRGSP attains a linear convergence rate. Our experimental results on large-scale synthetic and real-world datasets verify that SRGSP outperforms state-of-the-art related methods for tackling various sparse representation problems. Moreover, we conduct many experiments on two real-world sparse representation applications such as image denoising and face recognition, and all the results also validate that our SRGSP algorithm obtains much better performance than other sparse representation learning optimization methods in terms of PSNR and recognition rates.

Keywords: sparse learning; hard thresholding; stochastic optimization; variance reduction

1. Introduction

In recent years, sparse representation has been proved to be a useful approach to represent or compress high dimensional signals. Sparse representation algorithms have attracted many researchers in the fields of signal processing, image processing, medical imaging, machine learning, computer vision, pattern recognition, and so on [1]. In most applications, the unknown signal of interest is regarded as a sparse combination of a few columns from a given dictionary, and this problem is usually formulated as a sparsity constrained problem. Such sparse representation problems are common in the fields of image denoising, image inpainting, and face recognition or others such as [2–4].

Image denoising is a classical problem to improve image quality in computer vision. The aim of this problem is to recover the clean image \mathbf{x} from the noisy image $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where \mathbf{e} is additive white Gaussian noise in general [5]. It can be realized generally by the following three types of methods: transform domain [6], spatial filtering [7], and dictionary learning-based methods [8,9]. Note that the dictionary learning-based methods optimize the following model with the step of ℓ_0 -norm sparse coding:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{x}\|_0, \quad \text{s.t.}, \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \leq \varepsilon, \quad (1)$$

where \mathbf{x} represents the sparse coding of $y \in \mathbb{R}^n$ with a given error tolerance $\varepsilon > 0$, $\mathbf{D} \in \mathbb{R}^{n \times d}$ is a given dictionary, $\|\mathbf{x}\|_0$ denotes the number of nonzero entities of the vector \mathbf{x} , and $\|\mathbf{x}\|_2$ is the ℓ_2 -norm, i.e., $\|\mathbf{x}\|_2 = \sqrt{\sum_i \mathbf{x}_i^2}$. This is an ℓ_0 -norm constrained minimization problem, which can be solved via convex relaxation or greedy algorithms. The performance of the algorithms for finding a sparse representation solution has a big impact on denoised images. Similar effects can be found in face recognition tasks.

As a challenging application in computer vision and pattern recognition, face recognition has become a more complicated problem on account of illuminations, occlusions, expressions, and facial disguises. Many face recognition methods have been proposed, e.g., Eigenfaces [10] and Fisherfaces [11]. Wright et al. [12] proposed a sparse representation-based classification (SRC) method, which regards the training face images themselves as an overcomplete dictionary. It is possible for a testing face image to be represented by the atoms of the overcomplete dictionary, and thus we need to solve a sparse coding problem similar to Equation (1). Hence, there is the same question to image denoising. In this paper, our main aim is to find a more efficient algorithm for sparse representation on image denoising and face recognition tasks.

1.1. Stochastic Hard Thresholding Methods

So far, there have been many algorithms for pursuing sparse solutions using ℓ_p -norm ($0 \leq p \leq 1$) minimization [1]. Although convex relaxation algorithms are easy to find an optimal solution due to the convexity of ℓ_1 -norm, ℓ_1 -norm minimization has some limits as pointed out in [13] and sometime obtains worse empirical performance [14]. Thus, it is necessarily to solve the ℓ_0 -norm problem directly. In order to obtain the sparse solutions, many greedy algorithms [15–19] have been developed. Moreover, there are some hard thresholding-based methods, such as iterative hard thresholding [20], fast gradient hard thresholding pursuit [21], and gradient support pursuit (GraSP) [22]. All the methods have many successful applications for various real-world problems such as sparse vector and low-rank matrix recovery. However, the hard thresholding methods are deterministic optimization algorithms, they need to compute a full gradient using all training samples and have a high per-iteration complexity $O(nd)$, which makes them unsuitable for real-world large-scale data.

To address this issue, Nguyen et al. [23] proposed a stochastic gradient hard thresholding (SG-HT) algorithm, and introduced the idea of stochastic optimization into hard thresholding methods. It randomly selects one sample to optimize per-iteration and holds a much lower complexity. However, SG-HT cannot decrease the variance between the stochastic gradient and its full gradient. Li et al. [24] proposed a stochastic variance reduced gradient hard thresholding (SVRGHT) method, which uses the stochastic variance reduction gradient (SVRG) technique [25] as well as in [26]. With the help of variance reduction techniques, SVRGHT can obtain a faster convergence rate. More recently, there have been several stochastic hard thresholding algorithms using first-order or second-order information [27–33]. However, many stochastic algorithms such as SVRGHT have a hard thresholding operation in each iteration, whose computational complexity is relatively high $O(d \log(d))$ in general [34], especially for high-dimensional data. In addition, there are two main drawbacks for the thresholding methods. The first shortcoming is the optimization theoretical basis. That is, when the current iterate solution is not a minimizer of the function, moving from the solution in the direction of negative gradient of the function leads to the decrease in the value of this function. However, this assumption is not generally true when the hard thresholding operator $\mathcal{H}_s(\cdot)$ is applied to the current vector \mathbf{x}^t , which means that the gradient information has lost. It breaks the information of the current solution and may waste much computation to perform gradient descent. Secondly, the computational burden of hard thresholding operation is still linear with d , which can not be negligible. There exists an interesting question whether there is an algorithm to overcome these drawbacks. We answer this question affirmatively in theory and in practice.

1.2. Our Contributions

In this paper, we propose the first variance reduced stochastic recursive gradient method for sparse representation problems. In other words, we use the stochastic recursive gradient proposed in [35], which is suitable for solving non-convex problems, to optimize the non-convex sparse representation problem in this paper. In order to keep the gradient information of current iterate as suggested in [36], we perform lots of gradient descent steps, followed by a hard thresholding operation. We also construct the most relevant support on which minimization will be efficient. Therefore, this paper proposes a novel sparsity-constrained algorithm, called stochastic recursive gradient support pursuit (SRGSP). At each iteration in SRGSP, we first find the most relevant support set, minimize slackly over the support set by our stochastic recursive gradient solver, which satisfies a certain descent condition, and then perform hard thresholding on the updated model parameter. The main contributions and novelty of this paper are listed as follows:

- (1) It is non-trivial that we analyze the statistical estimation performance of SRGSP under mild assumptions, and the theoretical results show that SRGSP obtains a fast linear convergence rate.
- (2) Benefiting from less hard thresholding operations than existing algorithms such as SVRGHT, the average per-iteration cost of our algorithm is much lower ($O(d)$ for SRGSP vs. $O(d \log(d))$ for SVRGHT), which leads to faster convergence.
- (3) Moreover, less usage of hard thresholding operators to the current variable results in retain of gradient optimization information, which improves empirical performances. Stochastic recursive gradient support pursuit leads to a new trend to reduce the complexity of hard thresholding operation while maintaining or even improving the performance.
- (4) We also evaluate the empirical performance of our SRGSP method on sparse linear and logistic regression tasks as well as real-world applications such as image denoising and face recognition. Our experimental results show the efficiency and effectiveness of SRGSP.

The remainder of this paper is organized as follows. In Section 2, we introduce the related applications (i.e., image denoising and face recognition), and we propose our SRGSP algorithm in Section 3. The convergence analysis is provided in Section 4. In Section 5, many experimental results on both synthetic and real-world datasets verify the effectiveness of SRGSP, and the results of image denoising and face recognition further demonstrate the superiority of SRGSP against some state-of-the-art hard thresholding algorithms. Section 6 presents conclusions and future work.

2. Related Work

In this section, we start with a brief description of some related applications, in which sparse representation can play an important role.

2.1. Notation

In this paper, $\|\mathbf{x}\|_0$ denotes the number of nonzero entities of the vector \mathbf{x} , $\text{supp}(\mathbf{x})$ denotes the index set of nonzero entities of \mathbf{x} , and $\text{supp}(\mathbf{x}, s)$ is the index set of the top s entries of \mathbf{x} in terms of magnitude. In addition, we denote I^c the complement set of I and $\mathbf{x}|_I$ the restriction of vector \mathbf{x} to the rows indicated by indices in I . Furthermore, we denote $\mathbf{H}_{\mathcal{F}}(\cdot)$ the Hessian matrix of the function $\mathcal{F}(\cdot)$, and denote $\mathbb{E}(\cdot)$ the expectation.

2.2. Sparse Representation-based Image Denoising

In sparse representation, the clean images or signals can be approximated via a sparse combination of coefficients from a basis set, called dictionary. In this circumstance, denoising a patch vector y^j ,

which is extracted from the noisy image matrix, with a dictionary $\mathbf{D} \in \mathbb{R}^{n \times d}$ is regarded as solving the following sparsity constrained optimization problem:

$$\min_{\mathbf{x}} \mathcal{F}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \|y_i^j - \mathbf{D}_i \mathbf{x}\|_2^2, \text{ s.t., } \|\mathbf{x}\|_0 \leq s, \quad (2)$$

where $\mathbf{D}_i \mathbf{x}$ is an estimate of y_i^j , s is a sparsity constant, and y^j is the j -th patch of the noisy image y . There are many dictionary learning algorithms such as [8,37,38], which alternately update the dictionary with learned sparse iterate \mathbf{x} . Although these algorithms have demonstrated that learned dictionaries on noisy images or on a set of good quality images can achieve better performance than off-the-shelf ones such as [9], we here use the fixed overcomplete dictionary for verifying the property of our algorithm in sparse coding. The overcomplete dictionary \mathbf{D} , which means that the number of columns may be greater than the number of rows, can be obtained by the discrete cosine transform (DCT) [39] or its redundant version, as implemented in [38]. Since general images may be very large, current practices sparsely represent image patches rather than the full image.

In summary, we obtain an overcomplete dictionary matrix \mathbf{D} by DCT and then use ℓ_0 -norm constrained optimization algorithms to find an approximate solution of Equation (2) to restore the image.

2.3. Sparse Representation-based Face Recognition

Face recognition is an active research field in computer vision, and this task is to use k classes labeled training samples to classify the testing sample into the correct class. In this paper, we take our algorithm into the SRC framework [12] for face recognition. As SRC uses ℓ_1 -norm minimization to solve the sparse coding model, in this paper we use ℓ_0 -norm minimization instead, which can also obtain a sparse solution and this algorithm is provided in the Appendix A. In the SRC algorithm, an $l \times h$ gray facial training image is reshaped into a column vector $a_{r,u} \in \mathbb{R}^n$, i.e., $n = lh$. Then we construct the matrix $A_r = [a_{r,1}, a_{r,2}, \dots, a_{r,d_r}] \in \mathbb{R}^{n \times d_r}$ by using d_r training samples belonging to the r -th class. For each testing sample $y^r = [y_1^r, \dots, y_n^r]^T \in \mathbb{R}^{n \times 1}$ in the same class can be linear represented by the columns in A_r :

$$y^r = a_{r,1}x_{r,1} + a_{r,2}x_{r,2} + \dots + a_{r,d_r}x_{r,d_r}. \quad (3)$$

Here, $x_{r,1}, x_{r,2}, \dots, x_{r,d_r}$ are all scalars, which are the representation coefficients for y^r . Since the testing sample is unknown, then we consider all training samples of k classes and define a matrix A : $A = [A_1, A_2, \dots, A_k] \in \mathbb{R}^{n \times d}$. Therefore, the representation of a testing sample can be rewritten with respect to all the training samples as:

$$y^r = Ax_0, \quad (4)$$

where x_0 is a coefficient vector, whose nonzero elements only associated with the r -th class. In this paper, sparse representation with ℓ_0 -norm minimization can be used to solve the following sparsity constrained optimization problem in the SRC framework:

$$\min_{\mathbf{x}} \mathcal{F}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \|y_i^r - A_{row}^i \mathbf{x}\|_2^2, \text{ s.t., } \|\mathbf{x}\|_0 \leq s, \quad (5)$$

where s is the sparse constant, which implies the number of nonzero elements of \mathbf{x} and $A_{row}^i \in \mathbb{R}^{1 \times d}$ is the i -th row vector of A . Defining $\delta_p : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a selection function corresponding to the p -th class. Given a sparse vector $\mathbf{x} \in \mathbb{R}^d$ from Equation (5), $\delta_p(\mathbf{x}) \in \mathbb{R}^d$ is a new vector, whose nonzero elements are only associated with the p -th class. Then minimizing the following residual function can encode the identity of the sample y^r as follows:

$$\text{identity}(y^r) = \arg \min_p \{R_r(y) := \|y^r - A\delta_p(\mathbf{x})\|_2\}. \quad (6)$$

3. Our Stochastic Recursive Gradient Support Pursuit Method

In this section, we propose a novel stochastic recursive gradient support pursuit (SRGSP) method for sparsity constrained problems. Different from existing gradient support pursuit methods (e.g., GraSP [22]), SRGSP only requires to satisfy a certain constrictive condition in each iteration, and thus has a faster convergence speed in practice.

In recent years, many non-convex gradient support pursuit methods such as [20,22] have been proposed, and it has also been shown that they can have better performance than convex ℓ_1 -norm methods in certain circumstances. Most of the existing gradient support pursuit algorithms use deterministic optimization methods to minimize various sparse learning problems (e.g., Problem (2)). However, the per-iteration complexity of all these algorithms is $O(nd)$, which leads to slow convergence, especially for large-scale and high-dimensional problems. Inspired by GraSP [22], which is a well-known gradient support pursuit method, we propose an efficient stochastic recursive gradient support pursuit (SRGSP) algorithm to approximate the solution to Problem (2), as outlined in Algorithm 1.

Algorithm 1: Stochastic Recursive Gradient Support Pursuit (SRGSP)

Input: Sparsity level s , learning rate η , the numbers of outer-iterations and inter-iterations, T and J .

Initialize: $\hat{\mathbf{x}}^0$.

```

1: for  $t = 1, 2, \dots, T$  do
2:   Compute current gradient:  $\mathbf{g}^0 = \nabla \mathcal{F}(\hat{\mathbf{x}}^{t-1})$ ;
3:   Identify directions:  $\mathcal{Z} = \text{supp}(\mathbf{g}^0, 2s)$ ;
4:   Merge supports:  $\mathcal{T} = \mathcal{Z} \cup \text{supp}(\hat{\mathbf{x}}^{t-1})$ ;
5:   Initialization:  $\mathbf{z}^0 = \hat{\mathbf{x}}^{t-1}$ ,  $\mathbf{z}^1 = \mathbf{z}^0 - \eta \mathbf{g}^0$ ;
6:   for  $j = 1, 2, \dots, J$  do
7:     Randomly pick  $i_j \in \{1, 2, \dots, n\}$ ;
8:      $\mathbf{g}^j = \nabla f_{i_j}(\mathbf{z}^j) - \nabla f_{i_j}(\mathbf{z}^{j-1}) + \mathbf{g}^{j-1}$ ;
9:      $\mathbf{z}^{j+1} = \mathbf{z}^j - \eta \mathbf{g}^j$ ;
10:  end for
11:  Perform hard thresholding over  $\mathcal{T}$ :  $\hat{\mathbf{x}}^t = \mathcal{H}_s(\mathbf{z}^{J+1} |_{\mathcal{T}})$ ;
12: end for
Output:  $\hat{\mathbf{x}}^T$ .

```

At each iteration of Algorithm 1, we first compute the gradient of $\mathcal{F}(\cdot)$ at the current estimate, i.e., $\mathbf{g}^0 = \nabla \mathcal{F}(\hat{\mathbf{x}}^{t-1})$. Then we choose $2s$ coordinates of \mathbf{g}^0 that have the largest magnitude as the direction in which pursuing the minimization will be most effective, and denote their indices by \mathcal{Z} , where s is the sparsity constant. Merging the support of the current estimate with the $2s$ coordinates mentioned above, we can obtain the combined support, which is a set of at most $3s$ indices, i.e., $\mathcal{T} = \mathcal{Z} \cup \text{supp}(\hat{\mathbf{x}}^{t-1})$ (Some parameters used in this paper have already been defined in the second section). Over the current support set \mathcal{T} , we compute an estimate \mathbf{b} by using stochastic recursive gradient descent as the approximate solution to the problem (7).

The key difference between GraSP and SRGSP is that GraSP needs to yield the exact solution $\hat{\mathbf{b}}$ to the following minimization problem:

$$\min_{\mathbf{x}} \mathcal{F}(\mathbf{x}), \quad \text{s.t., } \mathbf{x} |_{\mathcal{T}^c} = \mathbf{0}, \quad (7)$$

where \mathcal{T}^c is the complement set of \mathcal{T} in line 4 of Algorithm 1, while our SRGSP method only requires a sub-solver (e.g., the iteration steps from Step 5 to Step 10 in Algorithm 1), which is to find an approximate solution \mathbf{b} to Problem (7) satisfying

$$\|\mathbf{b} - \hat{\mathbf{b}}\|_2 \leq c_1 \|\hat{\mathbf{x}}^{t-1} - \hat{\mathbf{b}}\|_2, \quad (8)$$

where $\hat{\mathbf{b}}$ is the exact solution to Equation (7), $\hat{\mathbf{x}}^{t-1}$ is the temporary result of last outer-iteration, $0 < c_1 < 1$ is an error bound constant, meaning that our algorithm can have a guaranteed decrease at each iteration, as shown in our convergence analysis in the next section. In other words, we can select other efficient solvers (e.g., SVRG [25], VR-SGD [32] and their accelerated variants [40,41]) for the proposed framework, as long as the solvers satisfy the certain constrictive condition in Equation (8). Since stochastic recursive gradient descent in [42,43] has been proved to have a faster convergence rate than other stochastic gradient operators such as SVRG [25] for solving non-convex optimization problems, we choose the former as our solver rather than SVRG as in [24]. When a fully deterministic optimization method is used as a sub-solver in Algorithm 1 for solving Problem (7), GraSP can be viewed as a special case of SRGSP.

In our experiments, we usually set $J = 2n$ as the number of iterations similar to the original SARA algorithm [35]. Within each inner-loop of Algorithm 1, our main update rules are as follows:

$$\begin{aligned} \mathbf{g}^j &= \nabla f_{i_j}(\mathbf{z}^j) - \nabla f_{i_j}(\mathbf{z}^{j-1}) + \mathbf{g}^{j-1}, \\ \mathbf{z}^{j+1} &= \mathbf{z}^j - \eta \mathbf{g}^j. \end{aligned}$$

Note that \mathbf{g}^j is the stochastic recursive gradient, which is first proposed in [35]. That is, our SRGSP algorithm updates \mathbf{g}^j using the accumulated stochastic information, which has the advantage of accelerating convergence naturally. The parameter $\hat{\mathbf{x}}^t$ is then updated using the hard thresholding operator, which keeps the largest s terms of the intermediate estimate \mathbf{b} . This step makes $\hat{\mathbf{x}}^t$ as the best s -term approximation of the estimate \mathbf{b} . The hard thresholding operator is defined as follows:

$$[\mathcal{H}_s(\mathbf{x})]_i = \begin{cases} x_i, & \text{if } i \in \text{supp}(\mathbf{x}, s), \\ 0, & \text{otherwise,} \end{cases}$$

where x_i is the i -th coordinate value of the vector \mathbf{x} .

Assumption 1. *The solution to the sub-problem (7) is unique.*

From the above analysis, we can find that our SRGSP algorithm uses a hard thresholding operation after lots of stochastic recursive gradient updates, while existing stochastic algorithms such as SVRGHT [24] perform hard thresholding in each inner-iteration, which is very time consuming for high-dimensional problems.

4. Convergence Analysis

In this section, we provide the convergence analysis of our SRGSP algorithm.

4.1. Convergence Property of Our Sub-solver

In this part, we consider the convergence property of our sub-solver in Algorithm 1, that is, our sub-solver can satisfy the descent condition in Equation (8). As most of the algorithms available in the community provide the bound for $\|\mathcal{F}(\mathbf{b}) - \mathcal{F}(\hat{\mathbf{b}})\|_2$, our convergence analysis requires other structures in $\mathcal{F}(\cdot)$ to obtain a bound for $\|\mathbf{b} - \hat{\mathbf{b}}\|_2$. Therefore, we would like to introduce the following insightful summary of the structures of $\mathcal{F}(\cdot)$ [44].

Lemma 1. Let $\mathcal{F}(\cdot)$ be a function with a Lipschitz-continuous gradient, the following implications hold:

$$(SC) \rightarrow (ESC) \rightarrow (WSC) \rightarrow (PL) \rightarrow (QG),$$

where SC means Strong Convexity, ESC means Essential Strong Convexity, WSC means Weak Strong Convexity, PL means Polyak–Lojasiewicz, and QG means Quadratic Growth. For their definitions, we would refer the reader to [44]. If we further assume that $\mathcal{F}(\cdot)$ is convex, then we have $(PL) \equiv (QG)$.

These results show that QG is the weakest assumption. Next, we prove that our sub-solver satisfies the descent condition in Equation (8).

Theorem 1. Suppose $\mathcal{F}(\cdot)$ satisfies the QG-condition with the parameter ρ and is Lipschitz continuous with the parameter L . Assume that the number of inter-iterations, J , is sufficiently large, our sub-solver has the following expected convergence property:

$$\mathbb{E}[\mathcal{F}(\mathbf{b}) - \mathcal{F}(\hat{\mathbf{b}})] \leq c_2[\mathcal{F}(\hat{\mathbf{x}}^{t-1}) - \mathcal{F}(\hat{\mathbf{b}})], \quad (9)$$

where $0 < c_2 < 1$ is a constant, and then we have

$$\|\mathbf{b} - \hat{\mathbf{b}}\|_2 \leq \frac{2c_2L}{\rho} \|\hat{\mathbf{x}}^{t-1} - \hat{\mathbf{b}}\|_2.$$

The detailed proofs of Theorem 1 and the theorem below are provided in the Supplementary Material. Similar to the linear convergence analysis of SARAH for solving convex problems in [35], our sub-solver can exhibit expected descent in the objective function value, as shown in Equation (9). If our sub-solver with a sufficiently large number of inter-iterations, then c_2 can be a very small constant. According to Theorem 1, one can easily verify that our sub-solver can satisfy the constrictive condition in Equation (8) when $\frac{2c_2L}{\rho} \leq c_1$. That is, our sub-solver with a sufficiently large number of inter-iterations can satisfy the constrictive condition in Algorithm 1.

4.2. Convergence Property of SRGSP

Before giving our main convergence result, we first present some important definitions.

Definition 1 (Stable Restricted Hessian). Suppose that $\mathcal{F}(\cdot)$ is a twice continuously differentiable function, and its Hessian matrix is denoted by $\mathbf{H}_{\mathcal{F}}(\cdot)$. For a given positive integer k , let

$$A_k(\mathbf{u}) = \sup_{|\text{supp}(\mathbf{u}) \cup \text{supp}(\mathbf{v})| \leq k, \|\mathbf{v}\|_2=1} \mathbf{v}^T \mathbf{H}_{\mathcal{F}}(\mathbf{u}) \mathbf{v}, \quad (10)$$

$$B_k(\mathbf{u}) = \inf_{|\text{supp}(\mathbf{u}) \cup \text{supp}(\mathbf{v})| \leq k, \|\mathbf{v}\|_2=1} \mathbf{v}^T \mathbf{H}_{\mathcal{F}}(\mathbf{u}) \mathbf{v}, \quad (11)$$

for all k -sparse vectors \mathbf{u} . Then $\mathcal{F}(\cdot)$ is said to have a Stable Hessian Property (SRH) with constant μ_k , or in short μ_k -SRH, if $1 \leq \frac{A_k(\mathbf{u})}{B_k(\mathbf{u})} \leq \mu_k$.

This definition shows that the SRH condition is similar to various forms of Restricted Strong Convexity (RSC) used in the performance analysis of existing sparsity constrained algorithms [22]. Note that this property is suitable for smooth loss functions, and there are a broad family of loss functions that have Lipschitz-continuous gradients.

Theorem 2. Let $\mathcal{F}(\cdot)$ be a twice continuously differentiable function that has μ_{4s} -SRH with $\mu_{4s} < \sqrt{2}$, and satisfies Assumption 1. For some $\epsilon > 0$, we have $\epsilon < B_{4s}(\mathbf{u})$ for all $4s$ -sparse \mathbf{u} , and $\{\hat{\mathbf{x}}^t\}$ is a sequence generated by Algorithm 1. Then we have

$$\|\hat{\mathbf{x}}^t - \mathbf{x}^*\|_2 \leq \delta^t \|\hat{\mathbf{x}}^0 - \mathbf{x}^*\|_2 + [1 - \delta^t] \frac{(1 + c_1)(2\mu_{4s} + 4)}{\epsilon(1 - \delta)} \|\nabla \mathcal{F}(\mathbf{x}^*)|_{\mathcal{I}}\|_2,$$

where $\delta := (1 + c_1)(\mu_{4s}^2 - 1) + 2c_1 < 1$, and \mathcal{I} is the position of the $3s$ largest entries of $\nabla \mathcal{F}(\mathbf{x}^*)$ in magnitude.

As discussed above, our sub-solver with a sufficiently large number of inter-iterations can satisfy the constrictive condition in Equation (8) with a very small constant c_1 , which makes $\delta < 1$ hold. Then we also have $0 < c_1 < \frac{2 - \mu_{4s}^2}{1 + \mu_{4s}^2}$. This implies that our sub-solver has to achieve a certain accuracy for the theorem to work. Theorem 2 suggests that our proposed algorithm achieves a linear convergence rate. This error bound consists of two terms, where the first term corresponds to the optimization error and the second term corresponds to the statistical error. After sufficient iterations, the second term will approach to zero. Therefore, our algorithm can always converge to the unknown true parameter \mathbf{x}^* with increasing of iterations.

5. Experimental Results

In this section, we evaluate the performance of our SRGSP method on synthetic and real-world large-scale datasets. Moreover, we apply SRGSP to tackle various sparse representation problems including image denoising and face recognition tasks. In this work, we only use the two real-world applications to illustrate the excellent performance of our SRGSP algorithm against other sparse learning optimization methods including GraSP [22], SG-HT [23], SVRGHT [24], and loopless semi-stochastic gradient descent with less hard thresholding (LSSG-HT) [34].

5.1. Baseline Methods

We compared the proposed algorithm (i.e., SRGSP) with four state-of-the-art algorithms: gradient support pursuit (GraSP) [22], stochastic gradient descent with hard thresholding (SG-HT) [23], stochastic variance reduced gradient with hard thresholding (SVRGHT) [24] and loopless semi-stochastic gradient descent with less hard thresholding (LSSG-HT) [34].

5.2. Synthetic Data

We generated a synthetic matrix A with size $n \times d$, each row of which is drawn independently from a d -dimensional Gaussian distribution with mean 0 and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. The response vector was generated from the linear model $y = A\mathbf{x}^* + \mathbf{e}$, where $\mathbf{x}^* \in \mathbb{R}^d$ is the s^* -sparse coefficient vector, and the noise \mathbf{e} was generated from a multivariate normal distribution $N(0, \sigma^2 I)$ with $\sigma^2 = 0.01$. The nonzero entries in \mathbf{x}^* were sampled independently from a uniform distribution over the interval $[-1, 1]$. For the experiments, we constructed two synthetic data: (1) $n = 2500$, $d = 5000$, $s^* = 250$, $\Sigma = I$; (2) $n = 5000$, $d = 10,000$, $s^* = 500$ and the diagonal entries of the covariance matrix Σ were set to 1, and the other entries were set to 0.1. The sparsity parameter s was set to $s = 1.2s^*$ for all the algorithms.

Figure 1 shows the performance (including the logarithm of the objective function values and the estimation error $\frac{\|\hat{\mathbf{x}}^t - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$) of all the algorithms on the synthetic data. All the results show that our algorithm converges significantly faster than the state-of-the-art methods in terms of function values and estimation error in all the settings. Although our SRGSP algorithm, SVRGHT and LSSG-HT have been theoretically proved to have a linear convergence rate, SRGSP consistently outperforms SVRGHT and LSSG-HT in terms of number of effective passes.

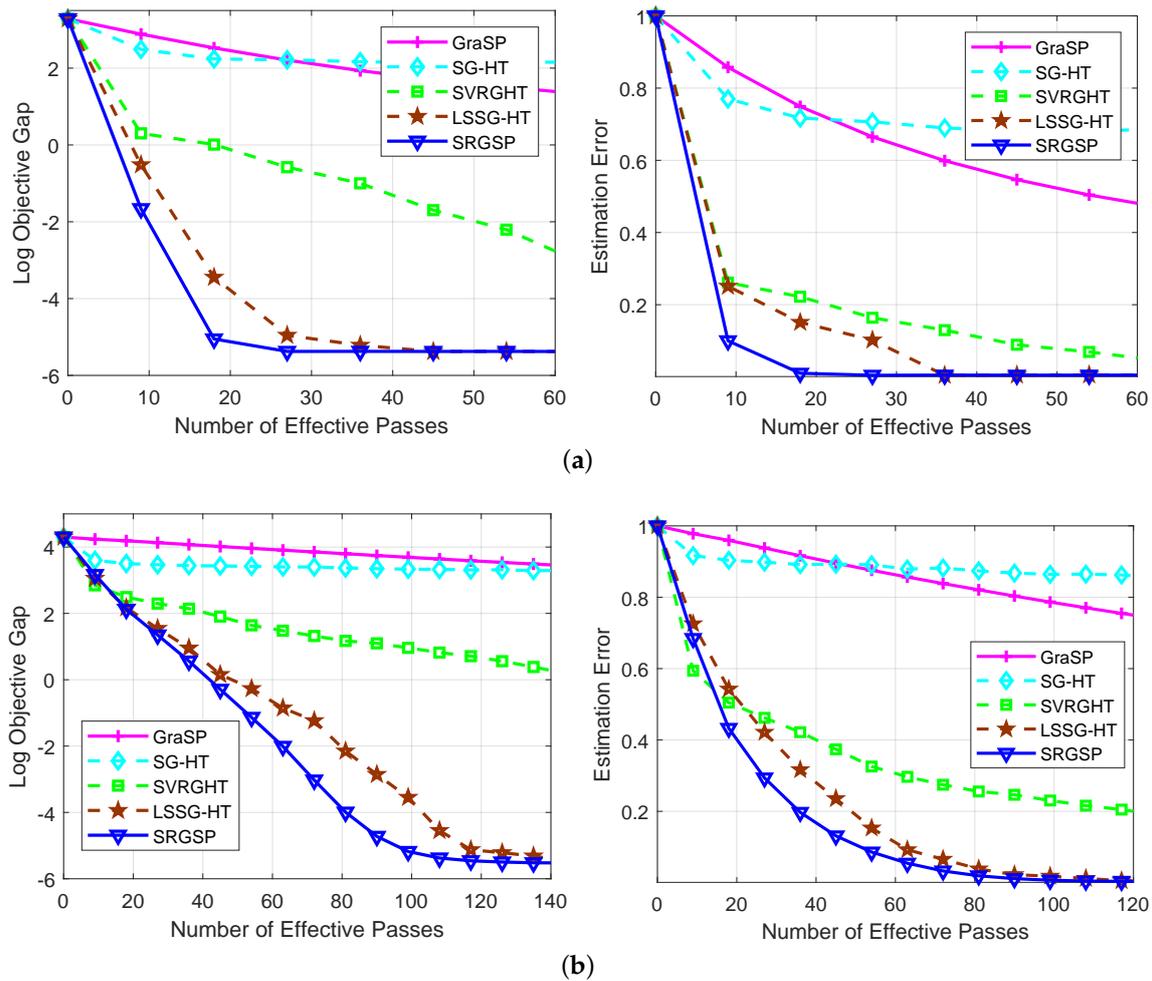


Figure 1. Comparison of gradient support pursuit (GraSP) [22], gradient support pursuit (SG-HT) [23], stochastic variance reduced gradient with hard thresholding (SVRGHT) [24], loopless semi-stochastic gradient descent with less hard thresholding (LSSG-HT) [34] and SRGSP for solving sparse linear regression problems on synthetic data. (a) $n = 2500, d = 5000, s^* = 250$; (b) $n = 5000, d = 10,000, s^* = 500$.

5.3. Real-world Data

In this subsection, we focus on the two real-world large-scale datasets: rcv1-train and real-sim, which can be downloaded from the LIBSVM website (<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>). In our experiments, we use the rcv1-train and real-sim datasets to evaluate the performance for linear regression, and the two datasets include 20,242 samples with 47,236 features and 72,309 samples with 20,958 features, respectively. Moreover, we choose $s = 200$ for the two datasets and compare all the algorithms in terms of logarithm objective value gap versus the number of effective passes and running time (in seconds).

Figure 2 illustrates the performance of our algorithm in terms of the logarithm of function gap (i.e., $\log(\|\mathcal{F}(\hat{x}^t) - \mathcal{F}(x^*)\|_2)$). More specifically, our SRGSP algorithm has a faster convergence rate than the four state-of-the-art sparsity constrained algorithms. In addition, SRGSP has the ability to jump out of a local minimum and can find a better solution, as shown in Figure 2b. Compared with SVRGHT, we can see that the results of our SRGSP in the first few iterations are similar to SVRGHT. However, due to lots of gradient updates followed by a hard thresholding, SRGSP can obtain a better solution, as discussed in [36]. This further verify the advantage of our SRGSP against other methods. On the other hand, our SRGSP algorithm can reach better solutions in much less CPU time than the other methods, including SVRGHT and LSSG-HT. Since SRGSP has one hard thresholding operation per iteration, while SVRGHT needs n operations (n is the number of samples) in each epoch,

and thus SVRGHT has a higher per-iteration complexity than SRGSP, especially for large-scale and high-dimensional data. Therefore, our SRGSP algorithm is very suitable for the large-scale non-convex sparsity constrained problem.

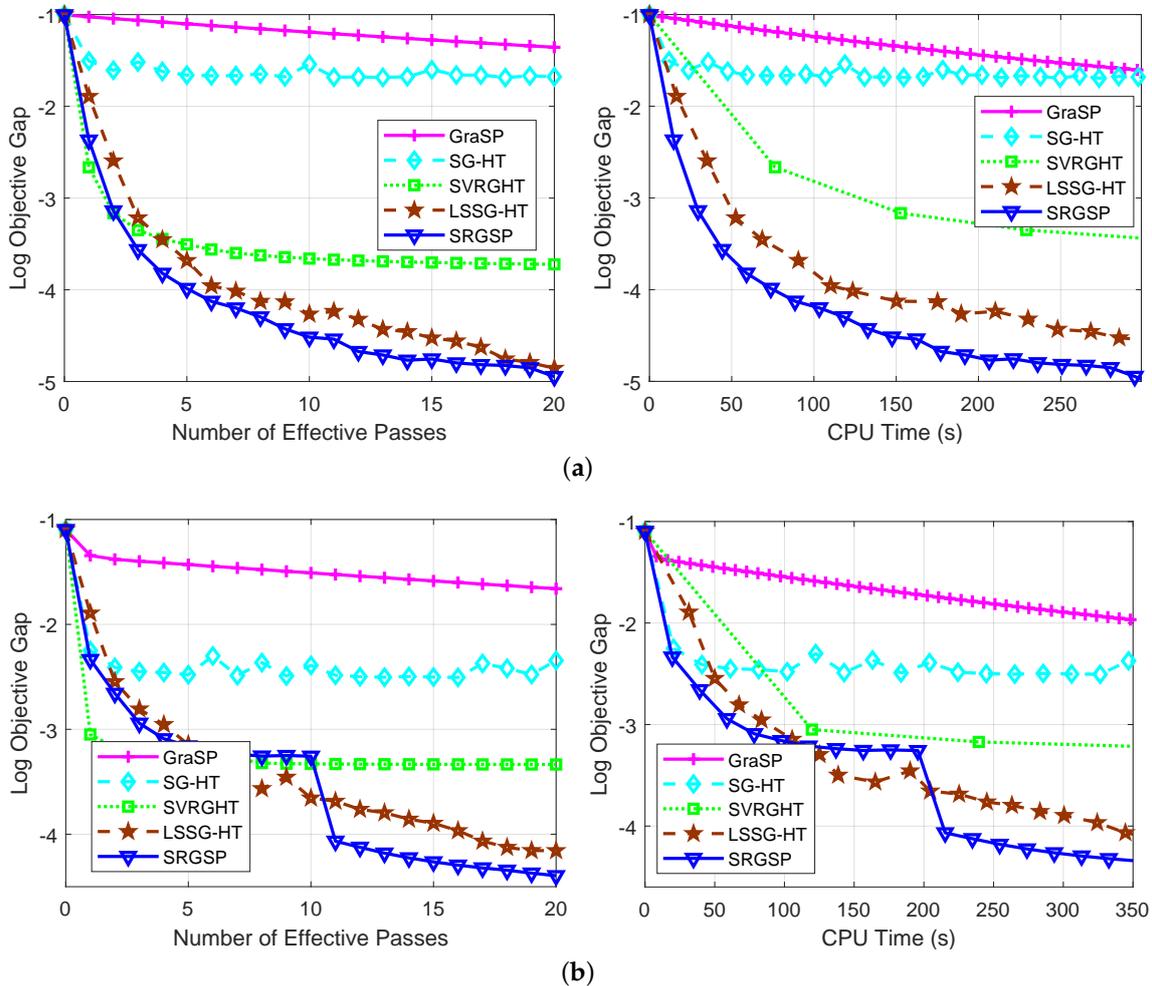


Figure 2. Comparison of GraSP [22], SG-HT [23], SVRGHT [24], LSSG-HT [34] and our SRGSP method for solving sparse linear regression problems. In each plot, the vertical axis shows the logarithm of the objective value minus the minimum, and the horizontal axis is the number of effective passes over data or running time (in seconds). (a) rcv1; (b) real-sim.

5.4. Image Denoising

In this subsection, we apply our SRGSP algorithm to image denoising tasks for evaluating its performance. First of all, the most important of image denoising is to find a suitable dictionary. In fact, the DCT seems like a reliable choice following the Guleryuz's work [45]. We use the overcomplete DCT dictionary in our experiments. Similar to [8], the error bound ε is set empirically as 1.15σ . The experiments are conducted on 6 standard benchmark images with synthetic white Gaussian noise. The sparsity level parameter in this experiment is set as $s = 10$ and the Gaussian random noise with zero mean and standard deviation $= \sigma$ is added to the standard images. The dictionary in the experiments is the fixed overcomplete DCT dictionary of size 64×256 , and is designed to deal with image patches of 8×8 pixels. The denoising processes are mainly concentrated on sparse coding of these patches using different sparsity constrained algorithms (e.g., GraSP, SG-HT, SVRGHT, LSSG-HT and SRGSP) and the classical greedy orthogonal matching pursuit (OMP) algorithm [46]. The parameters of \mathbf{x} are computed till the loss of Equation (1) lower than the error bound or the number of iterations larger than 32 (half of the row size of dictionary). When computing the sparse representative solution on

overlapping patches, all the algorithms have to evaluate the sparse coding solution of 62,001 patches for images of size 256×256 or 255,025 patches for images of size 512×512 . Then, the restoring patches were averaged in the same the procedure as in [8]. All the experiments are repeated 10 times, and the average results are reported.

Table 1 shows the results (including PSNR and SSIM) of all the algorithms at different noise levels, i.e., the values of σ vary from 5 to 55. As we can see from Table 1, our SRGSP algorithm can obtain higher PSNR and SSIM results than the other methods in all the settings, which indicates that the intrinsic low-dimensional structure can be found by our algorithm. Figure 3 shows the visual results of all the methods (i.e., SRGSP, LSSG-HT, SVRGHT, SG-HT, GraSP and OMP) on the cameraman image with $\sigma = 15$, where $s = 10$. It is clearly visible that the sky region of the cameraman image is well restored by SRGSP, while the results of other methods are not well recovered. Moreover, our SRGSP algorithm has a higher PSNR value of 29.08 dB, compared to 27.96 dB of SVRGHT and 27.32 dB of OMP. The SSIM results of OMP, SVRGHT and SRGSP are 0.6006, 0.8588, 0.8761, respectively. All the above results demonstrate the effectiveness of SRGSP for image denoising tasks.

Table 1. The denoising results (PSNR (dB)/SSIM) of all the methods including OMP [46], GraSP [22], SG-HT [23], SVRGHT [24], LSSG-HT [34] and our SRGSP method on 6 standard images at different noise levels from 5 to 55. The best performance is shown in bold.

| σ | Algorithms | Peppers | Cameraman | House | Man | Hill | Boat |
|----------|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 5 | SRGSP | 34.45 / 0.9259 | 34.12 / 0.9195 | 35.64 / 0.9322 | 34.15 / 0.9149 | 34.25 / 0.9132 | 34.68 / 0.9256 |
| | LSSG-HT | 33.56/0.9022 | 32.65/0.8932 | 34.12/0.9123 | 33.26/0.8865 | 33.35/0.8995 | 33.32/0.9203 |
| | SVRGHT | 33.95/0.9135 | 33.25/0.9065 | 34.01/0.9023 | 33.95/0.8997 | 33.39/0.9012 | 33.61/0.9165 |
| | SG-HT | 25.56/0.7801 | 26.62/0.7725 | 27.56/0.7835 | 27.65/0.7832 | 27.85/0.7532 | 25.89/0.7710 |
| | GraSP | 27.89/0.8832 | 24.72/0.8632 | 30.85/0.8857 | 28.25/0.7755 | 28.65/0.8278 | 26.91/0.7897 |
| | OMP | 34.02/0.8932 | 33.25/0.8706 | 34.23/0.8769 | 32.35/0.8562 | 33.35/0.8623 | 33.73/0.9143 |
| | 10 | SRGSP | 32.64 / 0.8942 | 32.05 / 0.8867 | 33.95 / 0.8932 | 32.13 / 0.8549 | 32.85 / 0.8535 |
| LSSG-HT | | 31.68/0.8822 | 30.22/0.7823 | 33.15/0.8734 | 31.56/0.8305 | 31.56/0.8462 | 31.98/0.8565 |
| SVRGHT | | 31.95/0.8835 | 29.56/0.7656 | 33.05/0.8721 | 31.35/0.8497 | 31.12/0.8342 | 31.56/0.8479 |
| SG-HT | | 24.48/0.7532 | 24.98/0.7272 | 26.85/0.7373 | 26.95/0.7326 | 26.89/0.7265 | 25.56/0.7235 |
| GraSP | | 27.56/0.8596 | 24.25/0.8323 | 30.15/0.8657 | 27.65/0.7651 | 28.26/0.7578 | 26.54/0.7589 |
| OMP | | 30.25/0.7656 | 29.56/0.7685 | 29.52/0.7685 | 28.36/0.7265 | 29.65/0.7552 | 29.89/0.7551 |
| 15 | | SRGSP | 30.57 / 0.8759 | 29.10 / 0.8707 | 32.81 / 0.8622 | 30.34 / 0.8249 | 30.65 / 0.7990 |
| | LSSG-HT | 29.95/0.8432 | 27.26/0.7102 | 32.12/0.8234 | 29.96/0.8105 | 30.24/0.7895 | 30.35/0.7956 |
| | SVRGHT | 30.35/0.8585 | 27.84/0.6927 | 32.26/0.8513 | 30.09/0.8197 | 30.34/0.7897 | 30.34/0.8079 |
| | SG-HT | 23.07/0.7311 | 22.92/0.6872 | 25.53/0.7073 | 26.12/0.7066 | 26.57/0.6721 | 24.89/0.6635 |
| | GraSP | 27.04/0.8449 | 23.46/0.8086 | 29.65/0.8357 | 26.97/0.7455 | 27.87/0.7178 | 26.11/0.7207 |
| | OMP | 27.76/0.6602 | 27.32/0.6006 | 27.80/0.5769 | 27.69/0.6504 | 27.70/0.6592 | 26.37/0.6051 |
| | 25 | SRGSP | 28.19 / 0.8232 | 27.37 / 0.8186 | 30.39 / 0.8224 | 28.10 / 0.7470 | 28.53 / 0.7164 |
| LSSG-HT | | 27.35/0.7785 | 26.35/0.5121 | 29.56/0.7806 | 27.85/0.7531 | 27.86/0.6842 | 27.96/0.7095 |
| SVRGHT | | 27.85/0.7606 | 26.85/0.5232 | 29.48/0.7797 | 27.92/0.7323 | 28.09/0.6998 | 27.82/0.7012 |
| SG-HT | | 22.34/0.6386 | 22.40/0.5810 | 24.68/0.5974 | 25.15/0.6030 | 25.57/0.5757 | 24.13/0.5711 |
| GraSP | | 26.12/0.8078 | 24.66/0.7715 | 28.50/0.8034 | 27.09/0.7188 | 27.00/0.6617 | 25.54/0.6748 |
| OMP | | 23.40/0.4696 | 23.23/0.4304 | 23.35/0.3838 | 23.30/0.4458 | 23.31/0.4457 | 21.91/0.4065 |
| 35 | | SRGSP | 26.55 / 0.7851 | 25.84 / 0.7674 | 28.62 / 0.7909 | 26.84 / 0.6966 | 27.29 / 0.6642 |
| | LSSG-HT | 25.16/0.7023 | 25.32/0.6812 | 27.65/0.7126 | 26.48/0.6610 | 26.56/0.6215 | 26.25/0.6126 |
| | SVRGHT | 25.94/0.7390 | 25.46/0.6907 | 27.46/0.7046 | 26.52/0.6627 | 26.85/0.6371 | 26.05/0.6025 |
| | SG-HT | 21.68/0.5514 | 21.77/0.4840 | 23.78/0.4916 | 24.28/0.5159 | 24.68/0.4950 | 23.37/0.4870 |
| | GraSP | 23.96/0.7546 | 22.91/0.7080 | 26.33/0.7549 | 25.67/0.6639 | 26.29/0.6269 | 25.02/0.6442 |
| | OMP | 20.46/0.3529 | 20.24/0.3264 | 20.40/0.2780 | 20.42/0.3215 | 20.38/0.3121 | 18.94/0.2882 |

Table 1. Cont.

| σ | Algorithms | Peppers | Camerman | House | Man | Hill | Boat |
|----------|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 45 | SRGSP | 25.26 / 0.7501 | 24.76 / 0.7258 | 27.53 / 0.7601 | 25.83 / 0.6575 | 26.40 / 0.6247 | 25.71 / 0.6519 |
| | LSSG-HT | 24.05/0.6502 | 24.12/0.6321 | 26.64/0.6532 | 25.32/0.5962 | 25.62/0.5933 | 25.51/0.5321 |
| | SVRGHT | 24.44/0.6722 | 24.24/0.6151 | 26.28/0.6375 | 25.46/0.6009 | 25.95/0.5807 | 24.71/0.5172 |
| | SG-HT | 21.05/0.4754 | 21.24/0.457 | 23.04/0.4120 | 23.38/0.4386 | 23.80/0.4207 | 22.63/0.4135 |
| | GraSP | 23.30/0.7264 | 22.70/0.6848 | 25.76/0.7330 | 25.03/0.6355 | 25.71/0.6014 | 24.43/0.6146 |
| | OMP | 18.27/0.2724 | 18.20/0.2604 | 18.24/0.2134 | 18.20/0.2384 | 18.18/0.2261 | 16.77/0.2176 |
| 55 | SRGSP | 24.28 / 0.7254 | 23.90 / 0.6946 | 26.28 / 0.7203 | 25.16 / 0.6271 | 25.74 / 0.5988 | 24.84 / 0.6179 |
| | LSSG-HT | 23.35/0.6212 | 23.01/0.5621 | 25.43/0.5963 | 24.52/0.5632 | 25.01/0.5423 | 24.32/0.4623 |
| | SVRGHT | 23.62/0.6183 | 23.22/0.5519 | 25.23/0.5708 | 24.74/0.5480 | 25.27/0.5349 | 23.61/0.4462 |
| | SG-HT | 20.51/0.4127 | 20.60/0.3484 | 22.25/0.3449 | 22.59/0.3774 | 22.98/0.3635 | 21.91/0.3546 |
| | GraSP | 22.86/0.7050 | 22.36/0.6646 | 25.04/0.7050 | 24.57/0.6136 | 25.30/0.5854 | 23.94/0.5918 |
| | OMP | 16.50/0.2203 | 16.43/0.2174 | 16.44/0.1665 | 16.48/0.1852 | 16.47/0.1720 | 15.02/0.1667 |

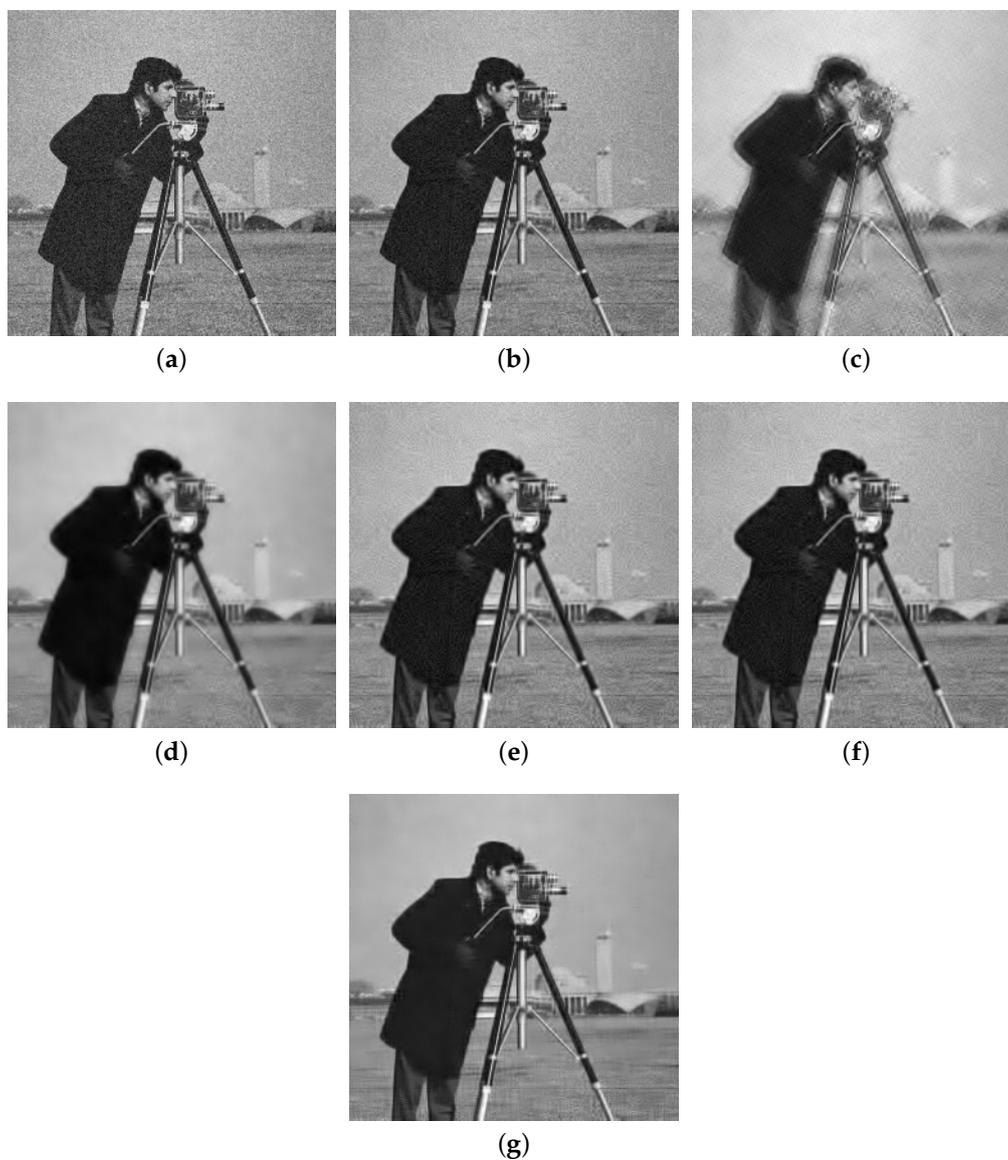


Figure 3. Comparison of the denoising results (PSNR/SSIM) of all the algorithms on the standard cameraman image with $\sigma = 15$, where $s = 10$. (a) Noise image ($\sigma = 15$); (b) OMP (27.62/0.6006); (c) SG-HT (22.90/0.7308); (d) GraSP (23.50/0.8453); (e) SVRGHT (27.96/0.8588); (f) LSSG-HT (27.26/0.8433); (g) SRGSP (29.08/0.8761).

Figures 4–6 show the denoising results of all the algorithms on the hill, pepper and boat images with different noise levels (e.g., $\sigma = 25, 35, 45$). We can see that our SRGSP algorithm consistently outperforms the other methods in terms of both PSNR and SSIM. Moreover, we give the following suggestion of empirical parameter-tuning for our SRGSP to obtain a good result. Based on all the experimental results, we find that in image denoising tasks, the error bound ϵ can be set empirically to 1.15σ for yielding a good result. For a general parameter setting, the number of outer-iteration T is set in the interval $[20, 30]$ and its inter-iteration is set to integer multiples of the number of samples.

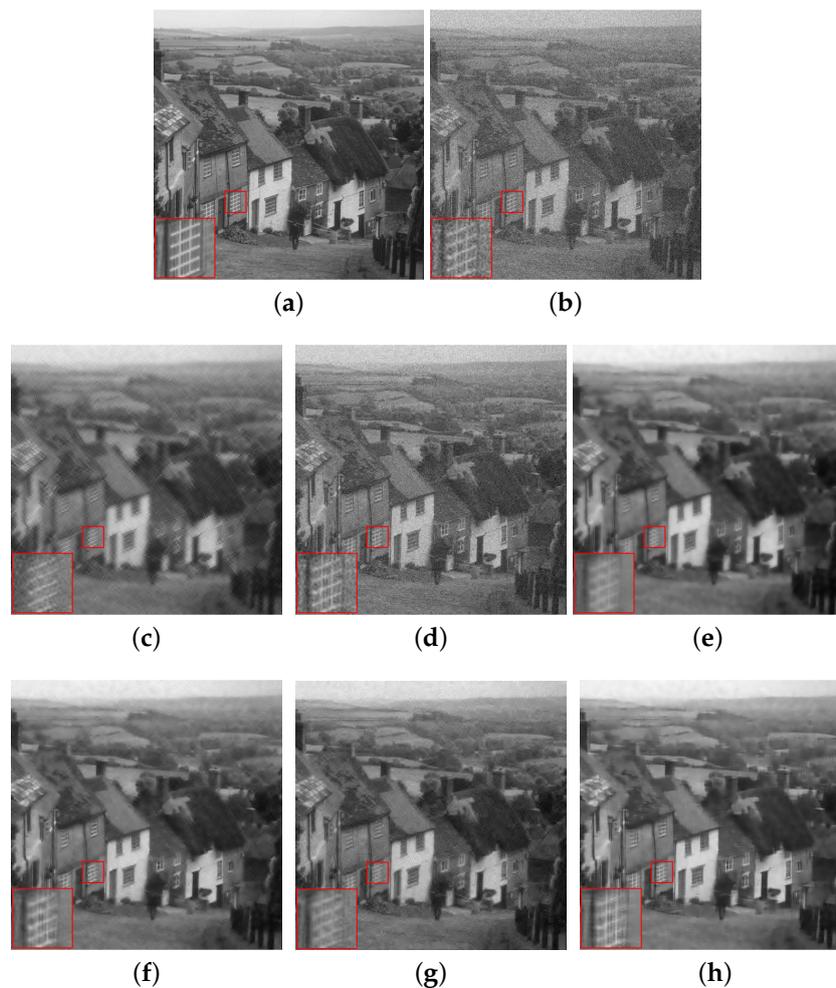


Figure 4. Comparison of the denoising results (PSNR/SSIM) of all the algorithms on the standard hill image with $\sigma = 25$, where $s = 10$. (a) Original image; (b) noise image ($\sigma = 25$); (c) SG-HT (25.58/0.5711); (d) OMP (21.90/0.4065); (e) GraSP (27.01/0.6748); (f) SVRGHT (28.16/0.7012); (g) LSSG-HT (27.86/0.7095); (h) SRGSP (28.53/0.7164).

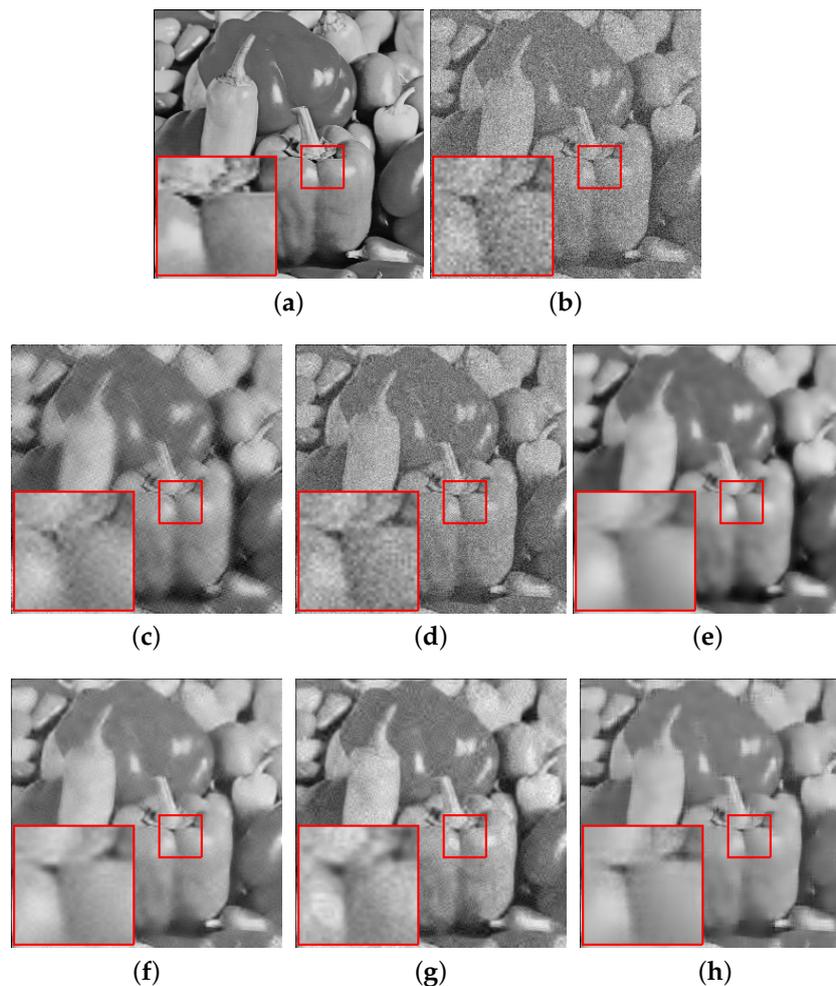


Figure 5. Comparison of the denoising results (PSNR/SSIM) of all the algorithms on the standard peppers image with $\sigma = 35$, where $s = 10$. (a) Original image; (b) noise image ($\sigma = 35$); (c) SG-HT (21.65/0.5514); (d) OMP (20.48/0.3529); (e) GraSP (23.91/0.7546); (f) SVRGHT (25.88/0.7390); (g) LSSG-HT (25.12/0.7023); (h) SRGSP (26.56/0.7851).

5.5. Face Recognition

In this subsection, we evaluate the performance of our SRGSP algorithm for face recognition on two real-world face datasets. More specifically, SRGSP is used as the solver in the sparse representation-based classification (SRC) framework [12]. We compare the recognition rates of SRGSP with those of some state-of-the-art sparsity constrained algorithms, such as GraSP, SG-HT, SVRGHT and LSSG-HT. In order to evaluate robustness of our algorithm, we manually add Gaussian noise to the face data.

5.5.1. Datasets

Although there are many datasets available for face recognition, we choose two common publicly datasets, i.e., the AR database [47] and the extended Yale B database [48]. The extended Yale B database contains 2414 frontal-face images of 38 people under different controlled lighting conditions [48]. For each individual, we randomly choose 26 images for training and 15 images for testing. The AR database contains over 4000 color images corresponding to 126 people's faces (70 Male and 56 Female). The images are obtained under different situation including different facial expressions, illumination conditions, and occlusions such as scarf or sunglasses. For simplicity, we randomly choose 100 objects, and each object has 15 images for training and 11 for testing. Note that the AR database may be

difficult for face recognition because there are more classes to recognize and few training samples for each object.

5.5.2. Experimental Setup

For each sparsity constrained algorithm, the sparsity level makes great difference to the solution of sparse representation, especially at different noise levels. Therefore, in order to approach the best performance of all these algorithms, we change the sparsity parameter within a certain range for each algorithm, i.e., $s \in [5, 10, 20, 30, 40, 50]$. Thus, we can make sure that all these algorithms achieve the best recognition rates in the parameter setting. In all settings of the experiments, the images are down-sampled to 32×32 pixels. As in [12], a series of processing operations are made to the above datasets. We first rescale the training matrix into $[0,1]$ for the convenience of adding noise, and then add Gaussian noise with zero mean and standard deviation σ . Finally, we normalize the columns of the training matrix to have unit ℓ_2 -norm.

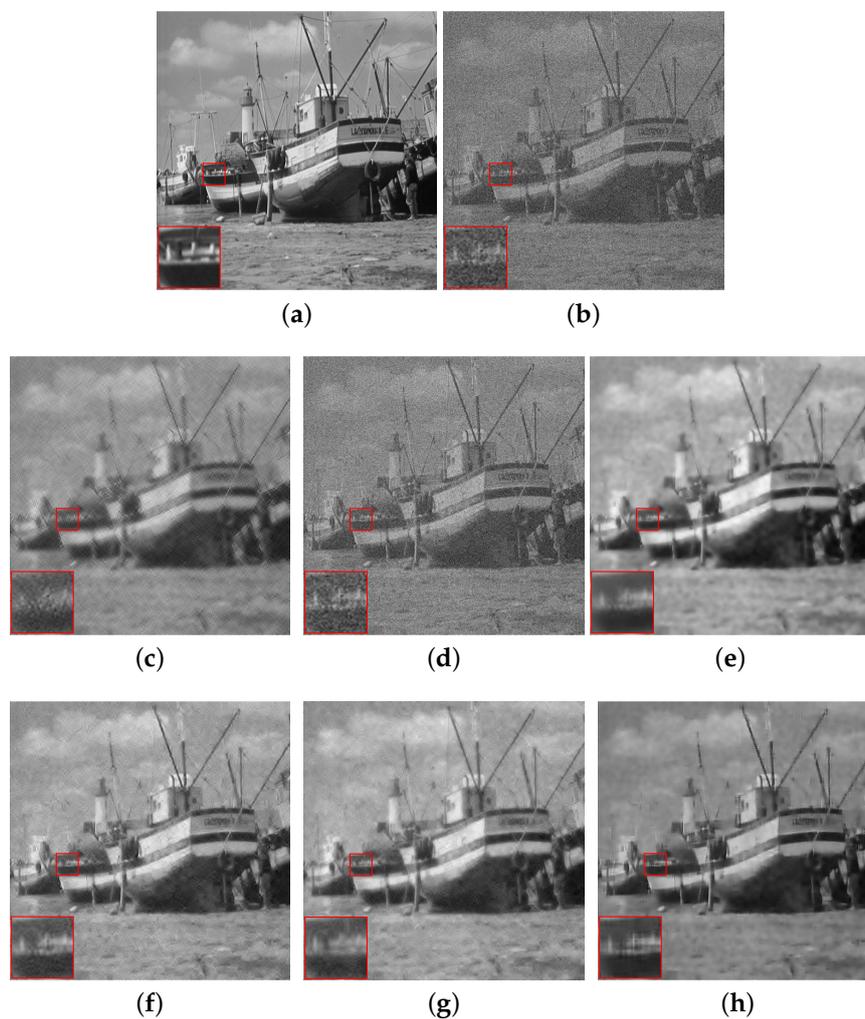


Figure 6. Comparison of the denoising results (PSNR/SSIM) of all the algorithms on the standard boat image with $\sigma = 45$, where $s = 10$. (a) Original image; (b) noise image ($\sigma = 45$); (c) SG-HT (22.62/0.4135); (d) OMP (16.77/0.2176); (e) GraSP (24.42/0.6146); (f) SVRGHT (24.68/0.5172); (g) LSSG-HT (25.51/0.5321); (h) SRGSP (25.68/0.6519).

5.5.3. Results on Real-World Face Data

In this part, we report the recognition rates of SRGSP on both the AR and the extended Yale B databases. Figure 7 shows the real testing images from the AR database with different Gaussian noise levels (e.g., $\sigma=0.25$ and $\sigma=0.5$). As we can see, it is challenging for humans to correctly recognize the face images under this situation. However, even in this extreme conditions, SRGSP achieves a high recognition rate with high probability.

In Figure 8, SVRGHT, LSSG-HT and SRGSP obtain much higher recognition rates than other two methods on both datasets, which verifies the superiority of variance reduction or recursive gradient techniques. Although at the very early iterations, SRGSP may have slightly lower recognize rates than SVRGHT, while with the increase of iterations, SRGSP can achieve the highest recognition rate among all the classifiers. Moreover, SRGSP is several times faster than SVRGHT due to less hard thresholding operations. For example, when $\sigma=0.25$ on the Yale B database, SRGSP obtains over 90% recognition rate within 50 s, while SVRGHT needs more CPU time to reach the same accuracy. In fact, in the same number of passes, SRGSP still achieves the highest recognition rate among the state-of-the-art hard thresholding algorithms, which demonstrates the effectiveness and efficiency of SRGSP. In the extreme situation of the AR database with the Gaussian noise level $\sigma=0.5$, SRGSP still achieves a higher recognition rate than the other methods. The results of recognition rates with respect to the number of passes on both datasets are provided in Figures 9 and 10, and further demonstrate the superiority of SRGSP.

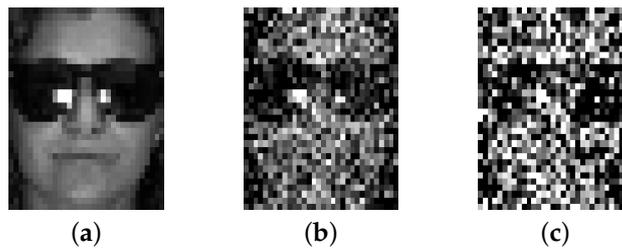


Figure 7. Examples of the AR face database with different levels of white Gaussian noise. (a) $\sigma=0$; (b) $\sigma=0.25$; (c) $\sigma=0.5$.

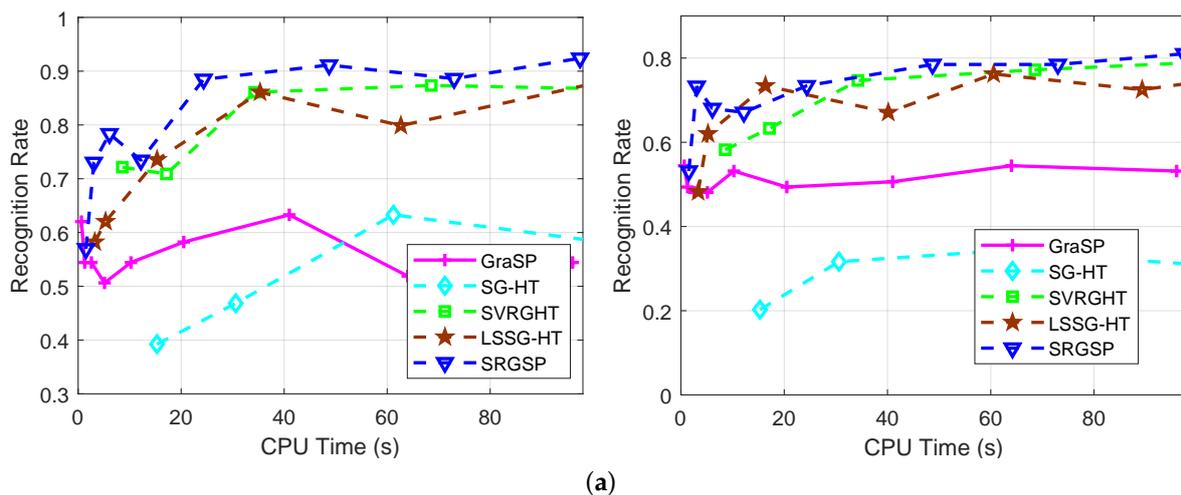


Figure 8. Cont.

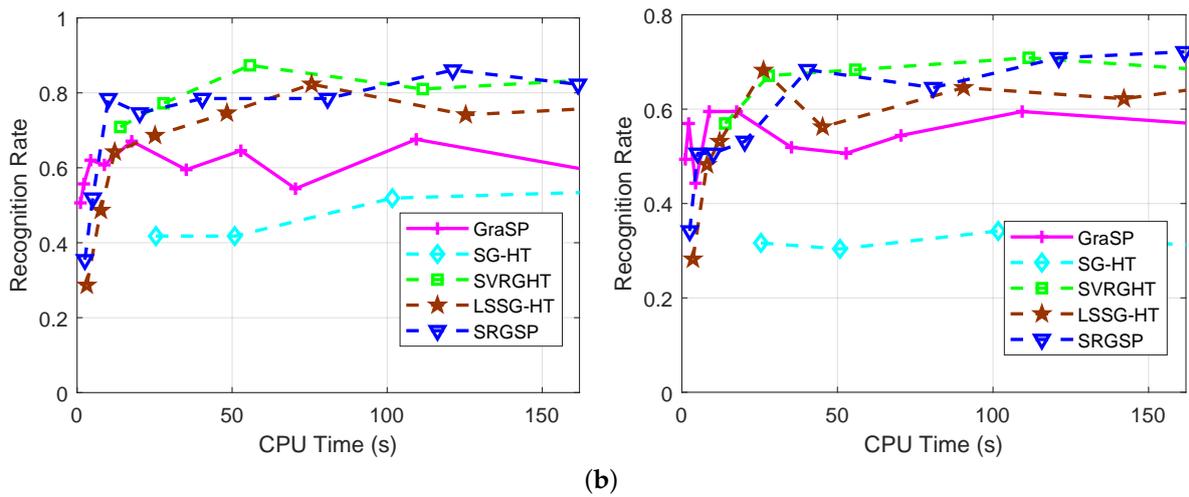


Figure 8. Recognition rates of all the algorithms on the extended Yale B (top) and AR (bottom) databases with different levels of Gaussian noise: $\sigma = 0.25$ and $\sigma = 0.5$. (a) The Yale B database with $\sigma = 0.25$ (left) and $\sigma = 0.5$ (right); (b) the AR database with $\sigma = 0.25$ (left) and $\sigma = 0.5$ (right).

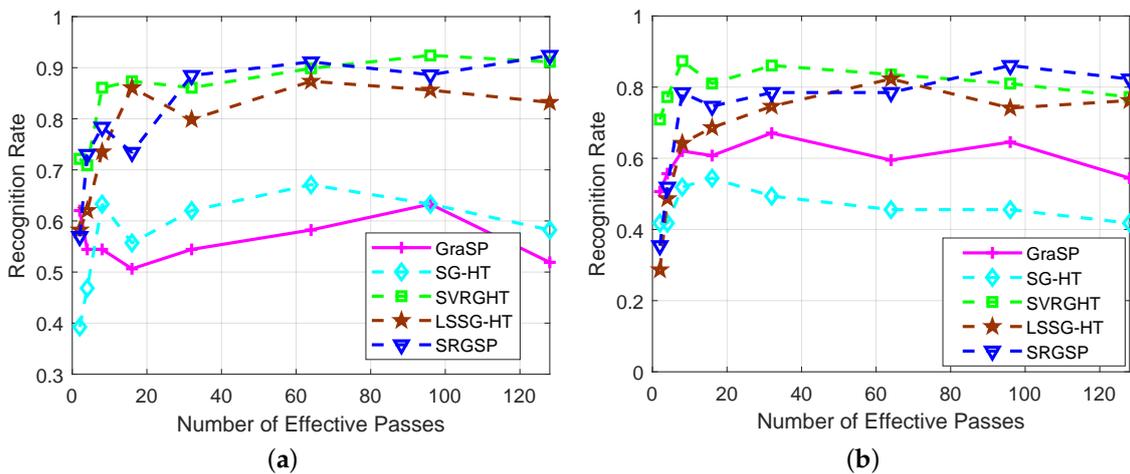


Figure 9. Comparison of recognition rates of all the algorithms on the extended Yale B and AR databases with the Gaussian noise level $\sigma = 0.25$. (a) The Yale B dataset; (b) the AR dataset.

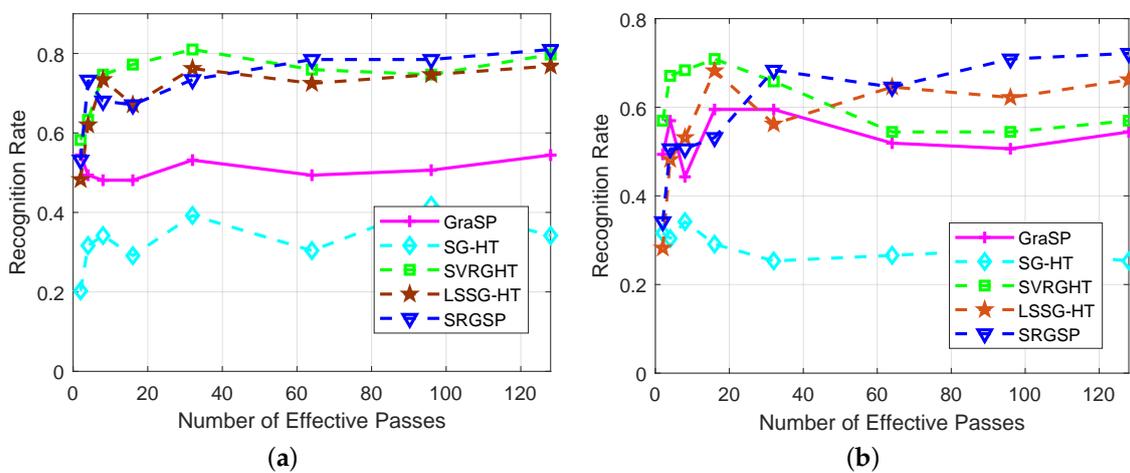


Figure 10. Comparison of recognition rates of all the algorithms on the extended Yale B and AR databases with the Gaussian noise level $\sigma = 0.5$. (a) The Yale B dataset; (b) The AR dataset.

6. Conclusions and Future Work

In this paper, we proposed a stochastic recursive gradient support pursuit (SRGSP) method for solving large-scale sparsity constrained optimization problems. We also provided the convergence analysis of SRGSP, which shows that SRGSP obtains a linear convergence rate. As existing hard thresholding-based algorithms need more thresholding operations, SRGSP just needs a hard thresholding operation in each epoch, and thus has a significantly per-iteration lower computational complexity, i.e., $O(d)$ vs. $O(d \log(d))$. Experimental results on synthetic and real large-scale datasets verified the effectiveness and efficiency of SRGSP.

Moreover, we also applied our SRGSP method to tackle image denoising and face recognition tasks, where sparse representation learning plays an important role. Our experimental results show that SRGSP outperforms other sparse representation methods in terms of PSNR and recognition rates. Note that for the image denoising application, the dictionary is the fixed overcomplete DCT dictionary. Inspired by some sophisticated methods such as K-SVD [38], we will iteratively update the dictionary in the future, which can further improve performance. In fact, there are many real-world sparse representation learning applications such as image super-resolution, image restoration, image classification and visual tracking. Therefore, we will apply our SRGSP method to address more applications in the future. In addition, our SRGSP algorithm can be extended to tackle low-rank matrix and tensor completion and recovery problems as in [49–51].

Author Contributions: Methodology and Formal analysis, F.S.; Software and Formal analysis, B.W.; Formal analysis, Y.L.; Investigation, H.L.; Visualization, S.W.; Review & editing, L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61976164, 61836009 and U1701267, and 61871310), the Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Science Foundation of Xidian University (Nos. 10251180018 and 10251180019), the National Science Basic Research Plan in Shaanxi Province of China (Nos. 2019JQ-657 and 2020JM-194), and the Key Special Project of China High Resolution Earth Observation System-Young Scholar Innovation Fund.

Acknowledgments: We thank all the reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this paper, we use $A = \{a_1, a_2, \dots, a_n\} \in \mathbb{R}^{d \times n}$ to denote the design matrix, $y = [y_1, y_2, \dots, y_d]^T \in \mathbb{R}^d$ to denote the response vector and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ to denote the model parameter. For the parameter vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_2$ denotes its ℓ_2 -norm, while $\|\mathbf{x}\|_1$ is the ℓ_1 -norm.

Sparse Representation-based Classification (SRC)

In this paper, we apply our algorithm into the SRC framework for face recognition tasks, as outlined in Algorithm A1.

Algorithm A1: Sparse Representation-based Classification

Input: A matrix of training samples $A = [A_1, A_2, \dots, A_k] \in \mathbb{R}^{d \times n}$ for k classes, sparsity parameter s , and a test sample $y \in \mathbb{R}^d$.

- 1: Normalize the columns of A to have unit ℓ_2 -norm;
- 2: Solve the ℓ_0 -minimization problem:

$$\hat{\mathbf{x}} = \arg \min \|y - A\mathbf{x}\|_2^2, \quad s.t., \|\mathbf{x}\|_0 \leq s;$$
- 3: Compute the residuals $r_i(y) = \|y - A\delta_i(\hat{\mathbf{x}})\|_2$ for $i = 1, \dots, k$;

Output: identity (y) = $\arg \min r_i(y)$.

References

1. Zhang, Z.; Xu, Y.; Yang, J.; Li, X.; Zhang, D. A survey of sparse representation: algorithms and applications. *IEEE Access* **2015**, *3*, 490–530.
2. Liu, S.; Hu, Q.; Li, P.; Zhao, J.; Wang, C.; Zhu, Z. Speckle Suppression Based on Sparse Representation with Non-Local Priors. *Remote Sens.* **2018**, *10*, 439.
3. Tu, B.; Zhang, X.; Kang, X.; Zhang, G.; Wang, J.; Wu, J. Hyperspectral Image Classification via Fusing Correlation Coefficient and Joint Sparse Representation. *IEEE Geoece Remote Sens. Lett.* **2018**, *15*, 340–344.
4. Liu, S.; Liu, M.; Li, P.; Zhao, J.; Zhu, Z.; Wang, X. SAR Image Denoising via Sparse Representation in Shearlet Domain Based on Continuous Cycle Spinning. *IEEE Trans. Geoece Remote Sens.* **2017**, *55*, 2985–2992.
5. Shao, L.; Yan, R.; Li, X.; Liu, Y. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. *IEEE Trans. Cybern.* **2014**, *44*, 1001–1013.
6. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K.O. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095.
7. Yan, R.; Shao, L.; Cvetkovic, S.D.; Klijn, J. Improved nonlocal means based on pre-classification and invariant block matching. *J. Disp. Technol.* **2012**, *8*, 212–218.
8. Elad, M.; Aharon, M. Image denoising via learned dictionaries and sparse representation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 895–900.
9. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; Zisserman, A. Non-local sparse models for image restoration. In Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 2272–2279.
10. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86.
11. Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720.
12. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227.
13. Candes, E.J.; Wakin, M.B.; Boyd, S.P. Enhancing sparsity by reweighted L_1 minimization. *J. Fourier Anal. Appl.* **2008**, *14*, 877–905.
14. Fan, J.; Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360.
15. Mallat, S.G.; Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415.
16. Pati, Y.C.; Rezaifar, R.; Krishnaprasad, P.S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–3 November 1993; pp. 40–44.
17. Needell, D.; Vershynin, R. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *arXiv* **2007**, arXiv:0712.1360
18. Dai, W.; Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory* **2009**, *55*, 2230–2249.
19. Needell, D.; Tropp, J.A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM* **2010**, *53*, 93–100.
20. Blumensath, T.; Davies, M. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.* **2009**, *27*, 265–274.
21. Yuan, X.; Li, P.; Zhang, T. Gradient Hard Thresholding Pursuit for Sparsity-Constrained Optimization. Available online: <http://proceedings.mlr.press/v32/yuan14.pdf> (accessed on 29 August 2020).
22. Bahmani, S.; Raj, B.; Boufounos, P.T. Greedy sparsity-constrained optimization. *J. Mach. Learn. Res.* **2013**, *14*, 807–841.
23. Nguyen, N.; Needell, D.; Woolf, T. Linear Convergence of Stochastic Iterative Greedy Algorithms with Sparse Constraints. *IEEE Trans. Inf. Theory* **2017**, *63*, 6869–6895.
24. Li, X.; Zhao, T.; Arora, R.; Liu, H.; Haupt, J. Stochastic Variance Reduced Optimization for Nonconvex Sparse Learning. Available online: <http://proceedings.mlr.press/v48/lid16.pdf> (accessed on 29 August 2020).

25. Johnson, R.; Zhang, T. Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. Available online: <http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction> (accessed on 29 August 2020).
26. Shen, J.; Li, P. A tight bound of hard thresholding. *J. Mach. Learn. Res.* **2017**, *18*, 7650–7691.
27. Chen, J.; Gu, Q. Accelerated Stochastic Block Coordinate Gradient Descent for Sparsity Constrained Nonconvex Optimization. In Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, New York, NY, USA, 25–29 June 2016.
28. Gao, H.; Huang, H. Stochastic Second-Order Method for Large-Scale Nonconvex Sparse Learning Models. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018), Stockholm, Sweden, 13–19 July 2018; pp. 2128–2134.
29. Chen, J.; Gu, Q. Fast newton hard thresholding pursuit for sparsity constrained nonconvex optimization. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 15–17 August 2017; ACM: New York, NY, USA, 2017; pp. 757–766.
30. Shang, F.; Liu, Y.; Cheng, J.; Zhuo, J. Fast stochastic variance reduced gradient method with momentum acceleration for machine learning. *arXiv* **2017**, arXiv:1703.07948.
31. Liang, G.; Tong, Q.; Zhu, C.; Bi, J. An Effective Hard Thresholding Method Based on Stochastic Variance Reduction for Nonconvex Sparse Learning. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 1585–1592.
32. Shang, F.; Zhou, K.; Liu, H.; Cheng, J.; Tsang, I.; Zhang, L.; Tao, D.; Jiao, L. VR-SGD: A Simple Stochastic Variance Reduction Method for Machine Learning. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 188–202.
33. Liu, Y.; Shang, F.; Liu, H.; Kong, L.; Jiao, L.; Lin, Z. Accelerated Variance Reduction Stochastic ADMM for Large-Scale Machine Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, doi:10.1109/TPAMI.2020.3000512.
34. Liu, X.; Wei, B.; Shang, F.; Liu, H. Loopless Semi-Stochastic Gradient Descent with Less Hard Thresholding for Sparse Learning. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; ACM: New York, NY, USA, 2019; pp. 881–890.
35. Nguyen, L.M.; Liu, J.; Scheinberg, K.; Takáč, M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2613–2621.
36. Zhao, Y.B. Optimal k -thresholding algorithms for sparse optimization problems. *SIAM J. Optim.* **2020**, *30*, 31–55.
37. Engan, K.; Rao, B.D.; Kreutz-Delgado, K. Frame design using FOCUSS with method of optimal directions (MOD). In Proceedings of the NORSIG, Oslo, Norway, September 1999; pp. 65–69.
38. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322.
39. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *100*, 90–93.
40. Zhou, K.; Shang, F.; Cheng, J. A Simple Stochastic Variance Reduced Algorithm with Fast Convergence Rates. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5975–5984.
41. Shang, F.; Jiao, L.; Zhou, K.; Cheng, J.; Ren, Y.; Jin, Y. ASVRG: Accelerated Proximal SVRG. In Proceedings of the Asian Conference on Machine Learning, Beijing, China, 14–16 November 2018; pp. 815–830.
42. Yuan, H.; Lian, X.; Li, C.J.; Liu, J.; Hu, W. Efficient Smooth Non-Convex Stochastic Compositional Optimization via Stochastic Recursive Gradient Descent. In *Advances in Neural Information Processing Systems*; NIPS: Vancouver, BC, Canada, 2019; pp. 6926–6935.
43. Zhou, P.; Yuan, X.T.; Yan, S.; Feng, J. Faster First-Order Methods for Stochastic Non-Convex Optimization on Riemannian Manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, doi:10.1109/TPAMI.2019.2933841.
44. Karimi, H.; Nutini, J.; Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 795–811.
45. Guleryuz, O.G. Nonlinear approximation based image recovery using adaptive sparse reconstructions. In Proceedings of the 2003 International Conference on Image Processing, Barcelona, Spain, 14–17 September 2003; pp. 713–716.

46. Tropp, J.A.; Gilbert, A.C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **2007**, *53*, 4655–4666.
47. Martinez, A.M. The AR face database. Available online: <https://ci.nii.ac.jp/naid/10011462458/> (accessed on 29 August 2020).
48. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660.
49. Shang, F.; Cheng, J.; Liu, Y.; Luo, Z.Q.; Lin, Z. Bilinear Factor Matrix Norm Minimization for Robust PCA: Algorithms and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2066–2080.
50. Liu, Y.; Shang, F.; Fan, W.; Cheng, J.; Cheng, H. Generalized Higher-Order Orthogonal Iteration for Tensor Decomposition and Completion. Available online: <http://papers.nips.cc/paper/5476-generalized-higher-order-orthogonal-iteration-for-tensor-decomposition-and-completion> (accessed on 29 August 2020).
51. Liu, Y.; Shang, F.; Fan, W.; Cheng, J.; Cheng, H. Generalized Higher Order Orthogonal Iteration for Tensor Learning and Decomposition. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2551–2563.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).