Sparse Learning with Stochastic Composite Optimization

Weizhong Zhang, Lijun Zhang, *Member, IEEE*, Zhongming Jin, Rong Jin, Deng Cai, *Member, IEEE*, Xuelong Li, *Fellow, IEEE*, Ronghua Liang, and Xiaofei He, *Senior Member, IEEE*

Abstract—In this paper, we study Stochastic Composite Optimization (SCO) for sparse learning that aims to learn a sparse solution from a composite function. Most of the recent SCO algorithms have already reached the optimal expected convergence rate $\mathcal{O}(1/\lambda T)$, but they often fail to deliver sparse solutions at the end either due to the limited sparsity regularization during stochastic optimization (SO) or due to the limitation in online-to-batch conversion. Even when the objective function is strongly convex, their high probability bounds can only attain $\mathcal{O}(\sqrt{\log(1/\delta)/T})$ with δ is the failure probability, which is much worse than the expected convergence rate. To address these limitations, we propose a simple yet effective two-phase Stochastic Composite Optimization scheme by adding a novel powerful sparse online-to-batch conversion to the general Stochastic Optimization algorithms. We further develop three concrete algorithms, OptimalSL, LastSL and AverageSL, directly under our scheme to prove the effectiveness of the proposed scheme. Both the theoretical analysis and the experiment results show that our methods can really outperform the existing methods at the ability of sparse learning and at the meantime we can improve the high probability bound to approximately $\mathcal{O}(\log(\log(T)/\delta)/\lambda T)$.

Index Terms—Sparse learning, stochastic optimization, stochastic composite optimization

1 Introduction

Stochastic Composite Optimization (SCO) methods have attracted considerable interests over the past few years [1], [2], [3], [4], [5]. They have already constituted an important class in Convex Optimization family, due to their inherently different flavor than traditional convex optimization and their desirable features in large scale machine learning, such as low computational complexity (per iteration) and low memory requirement. They aim to efficiently solve the following problem in a stochastic manner:

$$\min_{\mathbf{w}\in\mathcal{W}} \quad \phi(\mathbf{w}) = F(\mathbf{w}) + \Psi(\mathbf{w}), \tag{1}$$

where $F(\mathbf{w}) = \mathrm{E}_{z=(\mathbf{x},y)\sim\mathcal{P}_{XY}}[f(\mathbf{w},z)]$, $\mathcal{W}$ is a convex domain for the feasible solutions, $f(\mathbf{w},z)$ is a convex loss function in $\mathcal{W}$, $\Psi(\mathbf{w})$ is a regularizer that controls the complexity of the learned predictor $\mathbf{w}$, and $\mathcal{P}_{XY}$ is a joint distribution for the input pattern $\mathbf{x}$ and the output variable y. SCO methods are first-order methods [6, Section 1.1.2] and they are designed for the cases when calculating the gradient of $F(\mathbf{w})$ is intractable, such as when $\mathcal{P}_{XY}$ is unknown.

In this paper, we focus on one important special case of SCO, in which $\Psi(\mathbf{w})$ is a sparsity-inducing regularizer, such as the ℓ_1 norm for sparse vectors and the trace norm for low rank matrixes. The goal of this problem is similar to that of sparse learning or sparse online learning [7], which means only one training example is processed at each iteration.

Since SCO is included in the general Stochastic Optimization (SO) ([8], [9], [10]) family, SO methods for

$$\min_{\mathbf{w}\in\mathcal{W}} \quad \phi(\mathbf{w}) \tag{2}$$

can also be used for SCO problems. Unlike SCO methods, SO methods [8], [9], [11], [12] treat $\phi(\mathbf{w})$ as a whole and update the intermediate solutions based on the standard Stochastic Gradient Descent (SGD) method. That is to say, they obtain a stochastic gradient based on a randomly sampled training sample and update the solution by $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t\hat{\mathbf{g}}_t)$ at each iteration, where $\hat{\mathbf{g}}_t$ is the stochastic gradient of $\phi(\mathbf{w})$, and Π is the projection operator of $\mathcal{W}$. Since SO methods are not designed for sparse learning, they are short in delivering sparse solutions.

The key idea of most SCO methods [1], [2], [3], [13], [14], [15] is to introduce the Composite Gradient Mapping [16] into each iteration. Specifically, given the current intermediate solution $\mathbf{w}_t$, SCO methods update $\mathbf{w}_t$ by successively solving a series of problems as follows:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \ \mathcal{L}_t(\mathbf{w}) + \eta_t\Psi(\mathbf{w}), \tag{3}$$

- *W. Zhang, Z. Jin, D. Cai, and X. He are with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yuhang Tang Road, Hangzhou, Zhejiang 310058, China. E-mail: {zhangweizhongzju, jinzhongming888}@gmail.com, {dengcai, xiaofeihe}@cad.zju.edu.cn.*
- *L. Zhang is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, China. E-mail: zhanglj@lamda.nju.edu.cn.*
- *R. Jin is with the Alibaba Group, Seattle, WA 98057, USA. E-mail: jinrong.jr@alibaba-inc.com.*
- *X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transicent Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xian 710119, Shaanxi, P.R.China. E-mail: xuelong_li@opt.ac.cn.*
- *R. Liang is with the College of Information Engineering, Zhejiang University of Technology, No. 288 Liuhe Road, Hangzhou, Zhejiang 310058, China. E-mail: rhliang@zjut.edu.cn.*

Manuscript received 10 Jan. 2015; revised 1 Mar. 2016; accepted 24 May 2016. Date of publication 7 June 2016; date of current version 12 May 2017. Recommended for acceptance by H. Xu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2578323

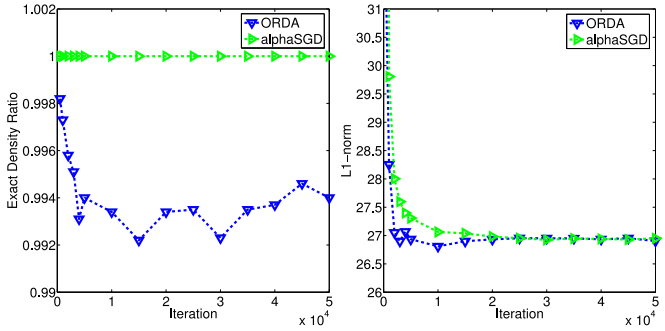


Fig. 1. The exact sparsity (i.e., the percentage of non-zero entries) (left) and ℓ_1 sparsity (right) of the solutions obtained by α-SGD and ORDA over iterations with $\lambda = 0.1, \rho = 0.1, \sigma_e^2 = 1$.

where $\mathcal{L}_t(\mathbf{w}) = (\mathbf{w} - \mathbf{w}_t)^T \hat{\mathbf{g}}_t + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2$. Here $\hat{\mathbf{g}}_t$ is a stochastic gradient of $F(\mathbf{w})$ at $\mathbf{w}_t$ and it is usually computed as $\hat{\mathbf{g}}_t = \partial f(\mathbf{w}_t, z_{t+1})$, where ∂ is the gradient (resp. subgradient) operator for differentiable functions (resp. general convex function), $z_{t+1} = (\mathbf{x}_{t+1}, y_{t+1})$ is a randomly sampled training sample. The main advantage of these approaches is that they can make use of the structure of the objective function $\phi(\mathbf{w})$, by which they can enforce the intermediate solutions to be what the regularizer $\Psi(\mathbf{w})$ prefers. For example, when $\Psi(\mathbf{w})$ is a sparse regularizer, the intermediate solutions obtained by (3) are likely to be sparse. Many approaches [3], [17] output the average of the intermediate solutions as the final solution. Compared to SO methods, it seems that the iterations of SCO are more complex, however, their computational complexities are comparable, since the subproblem (3) has a close form solution when $\Psi(\mathbf{w})$ is *simple* [4].

There is a potential problem in the existing SCO methods: even when $\Psi(\mathbf{w})$ is a sparsity-inducing regularizer, the final solution would not be sparse, especially not exactly sparse. As we mentioned above, in most of the traditional SCO methods, the final solution is obtained by taking the average of the intermediate solutions $\mathbf{w}_t$, a procedure which is referred to as online-to-batch conversion [18], [19]. Thus even if all the intermediate solutions are sparse, the final solution may be not sparse, at least not exactly sparse. To this end, some algorithms [2], [13] were designed recently by taking the solution in the last iteration as the final solution. They are always referred to as sparsity-preserving algorithms [13]. However, they are short in enforcing the final solution to be exactly sparse. The reason is that they have to reduce the magnitude of $\Psi(\mathbf{w})$, i.e., η_t, rapidly over the iterations to make the intermediate solutions converge to the optimal one, thus the effect of $\Psi(\mathbf{w})$ is weakened.

To demonstrate our point, we conduct a simple experiment on a synthetic dataset (see section 5.1 for details). In Fig. 1, we give the exact sparsity (left), i.e., the percentage of non-zero entries and ℓ_1 sparsity (right) curves of the solutions obtained by two well known approaches SO and SCO: α-SGD [8] that obtains the final solution by α-suffix averaging and ORDA [2] that takes the last solution as the final prediction model. It is clear that neither of them is able to obtain exactly sparse solutions, although the ℓ_1 sparsity of the solutions is improved over iterations.

In addition, we should note that a simple rounding method may not help to avoid the limitations above due to two reasons: 1) the solution $\mathbf{w}_t$ obtained by SCO is already subjected to the rounding effect of $\Psi(\mathbf{w})$ in the steps of Composite Gradient Mapping over the iterations, 2) due to the low convergence speed, SCO methods usually stop much earlier than they reach optimality. As a result, removing some of the small entries in the final solution which are important for the prediction task would make the rounded solution unreliable. This phenomenon will be demonstrated in our empirical study (Table 7).

In this paper, we devote ourselves to address the limitations described above. First, we propose a new two-phase scheme for SCO by introducing a novel sparse online-to-batch conversion step. Specifically, in phase one, we run a general SO algorithm to obtain an approximately optimal solution $\bar{\mathbf{w}}$. In phase two, we convert $\bar{\mathbf{w}}$ into an exactly sparse solution $\widetilde{\mathbf{w}}$ by using our sparse online-to-batch conversion procedure. And then, we design three concrete algorithms for SCO under this proposed scheme. Compared to the Composite Gradient Mapping in the existing SCO methods, our sparse online-to-batch conversion procedure is much more powerful at enforcing the solution to be exactly sparse. The best high probability bounds of our methods is approximately $\mathcal{O}(\log(\log(T)/\delta)/\lambda T)$, which is much better than those of the existing SCO methods.

2 Related Work

In this section, we will briefly review the recent work on SO, SCO and Sparse Online Learning. Just like most of the existing studies, we will concentrate on not only the sparse learning abilities of the approaches, but also the two forms of the convergence rates of these methods when $\phi(\mathbf{w})$ or $F(\mathbf{w})$ is λ-strongly convex, i.e., the expected convergence rate and the high probability bound (defined in Section 3).

2.1 Stochastic Optimization and Stochastic Composite Optimization

Most existing SO methods are derived from the standard SGD method, which computes the final solution by taking the average of the intermediate solutions to achieve faster convergence rate. There is a well known expected convergence rate $\mathcal{O}(\log(T)/\lambda T)$ for the standard SGD in [9]. The authors in [9] improved this rate to $\mathcal{O}(1/\lambda T)$ by designing a novel algorithm called "epoch gradient descent" (Epoch-GD), which was derived from SGD as well. Thus, the standard SGD seems to be suboptimal. In the recent investigation [8], the researchers show that when $\phi(\mathbf{w})$ is smooth, the standard SGD can also attain the optimal rate $\mathcal{O}(1/\lambda T)$, however, in non-smooth cases, the rate might really be $\mathcal{O}(\log(T)/\lambda T)$. A much simpler algorithm called "α-suffix average" was present in [8], which can reach the optimal convergence rate $\mathcal{O}(1/\lambda T)$. Compared to the standard SGD, the only difference is that α-suffix average takes the average of the last part of the intermediate solutions instead of all the solutions as the final output. Incidentally, in both Epoch-GD and α-suffix average, the high probability bounds are $\mathcal{O}(\log((\log T)/\delta)/\lambda T)$, where $0 < \delta < 1$ is the failure probability. Although both of them achieve the optimal convergence rate for strongly convex objective functions, they are not designed for sparse learning.

A multitude of SCO methods [2], [3], [13], [15] for Sparse Learning have been proposed based on the Composite Gradient Mapping in the recent years. In 2010, Xiao developed a algorithm called regularized dual average (RDA) [3] to obtain

sparse intermediate solutions. He proved that $\bar{\mathbf{w}}_T$ converges to the optimality $\mathbf{w}_*$ with the expected convergence rate $\mathcal{O}(\log(T)/(\lambda T))$ and high probability bound $\mathcal{O}(\sqrt{\log(1/\delta)/T})$, where $\bar{\mathbf{w}}_T = (\sum_{t=1}^{T}\mathbf{w}_t)/T$, which would not be sparse. However, there was no theoretical guarantee on the convergence of $\mathbf{w}_k$ in [3]. In 2011, Lin [13] proposed a sparse preserving algorithm SSG and gave the theoretical guarantee on $\mathbf{w}_k$ with the expected convergence rate $\mathcal{O}(1/(\lambda T))$ and the high probability bound $\mathcal{O}(1/(\delta\lambda T))$ (for λ-strongly convex $F(\mathbf{w})$) or $\mathcal{O}(\sqrt{\log(1/\delta)/T})$ (for general $F(\mathbf{w})$). However, $\mathcal{O}(1/(\delta\lambda T))$ is no better than $\mathcal{O}(\sqrt{\log(1/\delta)/T})$, due to its bad dependence on δ. The recent study [2] improved the expected convergence rate of [3] to $\mathcal{O}(1/(\lambda T))$ and strengthened its sparsity preserving ability by presenting a similar algorithm termed ORDA, which returns the last solution as the final prediction model. The high probability bound of ORDA is the same with that of SSG. So we can see that the high probability bounds of these SCO methods are all $\mathcal{O}(\sqrt{\log(1/\delta)/T})$ or $\mathcal{O}(1/(\delta\lambda T))$ even when $F(\mathbf{w})$ is λ-strongly convex, which are much worse than those of the recent SO methods. Although both SSG and ORDA avoid the problem of taking the average of intermediate solutions, the learned prediction model is likely to be approximately sparse instead of exactly sparse.

2.2 Sparse Online Learning

Sparse Online Learning focuses on obtaining a convergent sequence of sparse solutions that can minimize the learner's regret and it is closely related to the sparse recovery problem [20]. The main difference between them is that sparse recovery methods [21], [22] are designed for full gradients instead of stochastic gradients, and therefore are inapplicable to our case. Most of the existing approaches [7], [23] for Sparse Online Learning try to obtain sparse solutions by composite gradient mapping or other rounding processes. However, they always fail to learn an exactly sparse predictor due to the limitations of the online-to-batch conversion. More importantly, low-frequency features would tend to be truncated in most of the existing methods, which make it difficult to use these features for prediction [24]. In addition, although the goals of Sparse Online Learning and our problem are similar, their settings are slightly different, for example Online Sparse Learning does not require any distributional assumptions.

At last, we should point out that this journal paper is an extension of our own previous work [5].

3 Preliminary and Notation

Definitions. The following widely used definitions in SCO works are adopted to analyze our methods properly:

- Expected convergence rate: the convergence rate of $\mathrm{E}(\phi(\mathbf{w}_{out}) - \phi(\mathbf{w}_*))$, where $\mathbf{w}_{out}$ is the output of an algorithm after T iterations and $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}}\phi(\mathbf{w})$.
- High probability bound: a positive variable $\epsilon(T,\delta)$ satisfies $Prob(\phi(\mathbf{w}_{out}) - \phi(\mathbf{w}_*) > \epsilon(T,\delta)) < \delta$.

Notations and Assumptions. First, similar to most of the existing work, like [5], we assume that the loss function $f(\mathbf{w}, z)$ is G-Lipschitz continuous, λ-strongly convex and also L-smooth throughout this work. Then, our study is done under the SCO scheme, which means that we will randomly sample a training sample $z = (\mathbf{x}, y)$ at each iteration, and obtain a Stochastic Gradient $\hat{\mathbf{g}} = \partial f(\mathbf{w}, z)$ based on the sampled example. We update our solution $\widetilde{\mathbf{w}}$ after receiving each training sample z_t.

Goal. We want to find the predictor $\widetilde{\mathbf{w}}$ efficiently having the property that on one hand minimizes the objective $\phi(\mathbf{w})$ and on the other hand is as sparse as possible.

4 Our Stochastic Optimization Scheme for Sparse Learning

As we mentioned in the introduction section, our SCO scheme is comprised of two phases: 1) learn an approximately optimal solution $\bar{\mathbf{w}}$ by using a SO method, 2) refine $\bar{\mathbf{w}}$ into the final exactly sparse solution $\widetilde{\mathbf{w}}$ through a novel online-to-batch conversion procedure. In this section, we will present three specific approaches, OptimalSL, LastSL and AverageSL, under this scheme and analyze their performances respectively. The first one is a general algorithm, which can be combined with all the existing methods. The other two are based on the standard SGD method.

Before giving the detail steps of our methods, we would like to note here that the convergence of $\mathrm{E}(\phi(\mathbf{w}_{\mathrm{out}}) - \phi(\mathbf{w}_*))$ can not guarantee the quality of the solution of a single run, since we can construct a sequence of nonnegative random variables whose means converge to zero but variances go to infinity. The high probability bound is much more accurate and meaningful since it evaluates the reliability of an algorithm in a probabilistic way. And it can be used to evaluate the variance since the holding of the inequality $\mathrm{Var}(\phi(\mathbf{w}_{\mathrm{out}})) \leq \mathrm{E}((\phi(\mathbf{w}_{\mathrm{out}}) - \phi(\mathbf{w}_*))^2)$. For this reason and the space limitation, we only give the high probability bounds in our work.

Algorithm 1. Sparse Learning Based on Existing SO Methods

1: **Input:** strong convexity $\lambda > 0$, smoothness $L > 0$, tradeoff parameter $0 < \alpha < 1$, training examples $\{z_t = (\mathbf{x}_t, y_t)\}_{t=1}^{T}$, and a Stochastic Composite Optimization algorithm $\mathcal{A}$.
2: Run $\mathcal{A}$ with the first $(1-\alpha)T$ training examples to obtain approximately optimal solution $\bar{\mathbf{w}}_{1-\alpha}$, i.e., $\bar{\mathbf{w}}_{1-\alpha} = \mathcal{A}(\lambda, L, (1-\alpha)T)$.
3: // Sparse online-to-batch conversion:
4: Compute the average gradient at $\bar{\mathbf{w}}_{1-\alpha}$ using the remaining αT training examples

$$\bar{\mathbf{g}}_{1-\alpha}^{\alpha} = \frac{1}{\alpha T}\sum_{i=1+(1-\alpha)T}^{T}\nabla f(\bar{\mathbf{w}}_{(1-\alpha)}, z_i)$$

5: Compute the final solution $\widetilde{\mathbf{w}}$ as

$$\widetilde{\mathbf{w}} = \arg\min\langle\bar{\mathbf{g}}_{1-\alpha}^{\alpha}, \mathbf{w}\rangle + \frac{L}{2}\|\mathbf{w} - \bar{\mathbf{w}}_{1-\alpha}\|^2 + \Psi(\mathbf{w}) \quad (4)$$

6: **Return:** $\widetilde{\mathbf{w}}$

4.1 Sparse Learning Based on Existing SO Methods

Our first approach is a general algorithm that can make use of existing methods to get an approximately optimal solution and we named it OptimalSL. Its detailed steps are given in Algorithm 1. In phase one, we run an existing SO algorithm $\mathcal{A}$

to get an approximately solution $\bar{\mathbf{w}}_{1-\alpha}$ with $(1-\alpha)T$ training examples. Phase two is a novel sparse online-to-batch conversion. It first calculates the gradient of $F(\mathbf{w})$ at $\bar{\mathbf{w}}_{1-\alpha}$ based on the rest αT training examples. Then it refines $\bar{\mathbf{w}}_{1-\alpha}$ into $\widetilde{\mathbf{w}}$ by a composite gradient mapping (4), which has an explicit solution when the regularizer $\Psi(\mathbf{w})$ is simple, such as ℓ_1 norm.

As shown in Algorithm 1, there is a parameter α in our approach. It is introduced to balance SO and online-to-batch conversion. The mechanism of how it acts is given in the theoretical analysis below.

One important thing we should note here is that we use the original sparse regularizer $\Psi(\mathbf{w})$ without reducing its size in the composite gradient mapping (4), while in most existing SCO methods it vanishes rapidly over iterations. This difference would lead to an exactly sparse solution $\widetilde{\mathbf{w}}$. It is particularly clear when $\Psi(\mathbf{w}) = \beta\|\mathbf{w}\|_1$. If we note $\mathbf{v} = L\bar{\mathbf{w}}_{1-\alpha} - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}$, then the solution of (4) can be given by

$$[\widetilde{\mathbf{w}}]_i = \begin{cases} 0, & if\ |[\mathbf{v}]_i| < \beta \\ \frac{1}{L}[\mathbf{v} - \beta \mathrm{sgn}(\mathbf{v})]_i, & \text{else} \end{cases}$$

$[\widetilde{\mathbf{w}}]_i$ would be exactly 0 when $|[\mathbf{v}]_i| < \beta$, our point is thus confirmed.

Another important thing is that the online-to-batch conversion step is quit different from a simple rounding approach and the gradient $\bar{\mathbf{g}}^{\alpha}_{1-\alpha}$ is important to ensure that the final sparse solution $\widetilde{\mathbf{w}}$ also minimizes the objective function $\phi(\mathbf{w})$. This is justified by the theorem below.

Before giving the main theoretical bound, we require that the method $\mathcal{A}$ we used in our algorithm has the recent optimal high probability bound. In particular, we choose lemma 2 in [8] as our requirement:

Requirement 1. For any $\delta \in (0, 1/e)$ and $T \geq 4$, then it holds with probability at least $1-\delta$, that for any $t \leq T$, we have $\phi(\bar{\mathbf{w}}_{1-\alpha}) - \phi(\mathbf{w}_*) \leq \big(C_2 \log(\log((1-\alpha)T)/\delta)G^2\big) / \big(\lambda(1-\alpha)T\big)$ for some constant C_2.

Theorem 1. *Suppose the loss function $f(\mathbf{w}, z)$ is G-Lipschtiz continuous, λ-strongly convex and L-smooth. Assume $\mathcal{A}$ is an optimal SCO algorithm that satisfies our Requirement 1 above. Let $\delta \in (0, 1/e)$ and assume $T \geq 4$. Then it holds with probability at least $1-2\delta$ that*

$$\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \leq \frac{C_2 \log(\log((1-\alpha)T)/\delta)G^2}{\lambda(1-\alpha)T} + \frac{64G^2(\log(2/\delta))^2}{\alpha T}.$$

We need the following two lemmas from [9] and [25] respectively to derive Theorem 1,.

Lemma 1. *Let $f(\mathcal{W})$ be a λ-strongly convex function over the domain $\mathcal{W}$, and $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$. Then, for any $\mathbf{w} \in \mathcal{W}$, we have*

$$f(\mathbf{w}) - f(\mathbf{w}_*) \geq \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}^*\|^2.$$

Lemma 2. *Let $\mathcal{H}$ be a Hilbert Space and let ξ be a random variable on $(\mathcal{Z}, \rho)$ with values in $\mathcal{H}$. Assume $\|\xi\| \leq B < \infty$ almost surely. Let $\{\xi_i\}_{i=1}^m$ be independent random draws of ρ. For any $0 < \delta < 1$, with a probability at least $1-\delta$,*

$$\Big\|\frac{1}{m}\sum_{i=1}^{m}\big(\xi_i - \mathrm{E}[\xi_i]\big)\Big\| \leq \frac{4B}{\sqrt{m}}\log\frac{2}{\delta}.$$

Proof of Theorem 1.

First, due to the L-strongly convex of the function $h(\mathbf{w}) \triangleq \langle \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \mathbf{w}\rangle + \frac{L}{2}\|\mathbf{w} - \bar{\mathbf{w}}_{1-\alpha}\|^2 + \Psi(\mathbf{w})$ and the definition of $\widetilde{\mathbf{w}}$, by applying Lemma 1 to $\mathbf{h}(\mathbf{w})$, we can get:

$$\begin{aligned} &\langle \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \widetilde{\mathbf{w}}\rangle + \frac{L}{2}\|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\|^2 + \Psi(\widetilde{\mathbf{w}}) \\ &\quad \leq \langle \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \bar{\mathbf{w}}_{1-\alpha}\rangle + \Psi(\bar{\mathbf{w}}_{1-\alpha}) - \frac{L}{2}\|\bar{\mathbf{w}}_{1-\alpha} - \widetilde{\mathbf{w}}\|^2. \end{aligned} \tag{5}$$

Then, since $F(\mathbf{w})$ is L-smooth, we have:

$$\begin{aligned} \phi(\widetilde{\mathbf{w}}) &= F(\widetilde{\mathbf{w}}) + \Psi(\widetilde{\mathbf{w}}) \\ &\leq F(\bar{\mathbf{w}}_{1-\alpha}) + \langle \nabla F(\bar{\mathbf{w}}_{1-\alpha}), \widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\rangle \\ &\quad + \frac{L}{2}\|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\|^2 + \Psi(\widetilde{\mathbf{w}}) \\ &= F(\bar{\mathbf{w}}_{1-\alpha}) + \langle \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\rangle + \frac{L}{2}\|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\|^2 \\ &\quad + \Psi(\widetilde{\mathbf{w}}) + \langle \nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\rangle \\ &\overset{(5)}{\leq} F(\bar{\mathbf{w}}_{1-\alpha}) + \Psi(\bar{\mathbf{w}}_{1-\alpha}) - \frac{L}{2}\|\bar{\mathbf{w}}_{1-\alpha} - \widetilde{\mathbf{w}}\|^2 \\ &\quad + \langle \nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\rangle \\ &\leq F(\bar{\mathbf{w}}_{1-\alpha}) + \Psi(\bar{\mathbf{w}}_{1-\alpha}) - \frac{L}{2}\|\bar{\mathbf{w}}_{1-\alpha} - \widetilde{\mathbf{w}}\|^2 \\ &\quad + \frac{1}{2L}\|\nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}\|^2 + \frac{L}{2}\|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}_{1-\alpha}\|^2 \\ &= \phi(\bar{\mathbf{w}}_{1-\alpha}) + \frac{1}{2L}\|\nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}\|^2. \end{aligned}$$

Since $\|\nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}\|^2 = \|\frac{1}{\alpha T}\sum_{i=1}^{\alpha T}(\nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \hat{\mathbf{g}}(\bar{\mathbf{w}}_{1-\alpha}, i))\|^2$, by using lemma 2 under our G-Lipschitz assumption, we get, it holds with probability at least $1-\delta$ that

$$\|\nabla F(\bar{\mathbf{w}}_{1-\alpha}) - \bar{\mathbf{g}}^{\alpha}_{1-\alpha}\|^2 \leq \big(64G^2(\log(2/\delta))^2/(\alpha T)\big).$$

Finally, we have with probability at least $1-2\delta$:

$$\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \leq \frac{C_2 \log(\log((1-\alpha)T)/\delta)G^2}{\lambda(1-\alpha)T} + \frac{64G^2(\log\frac{2}{\delta})^2}{\alpha T}$$

□

Theorem 1 tells us that the high probability bound of our Algorithm 1 roughly equals to $\mathcal{O}(\log(\log(T)/\delta)/\lambda T)$, which is the same with the SO methods [8], [9] and much better than the recent SCO works [2], [13]. Just as we mentioned above, the parameter α indeed plays a balance role between the loss of $\mathcal{A}$ and the loss of sparse online-to-batch conversion: a small α will lead to a small error in SO but a large error in phase two.

4.2 Sparse Learning Based on the Last Solution

There is a potential drawback in Algorithm 1 that the training examples are underused, which is verified in Fig. 6. Only a portion of training examples are used by the SO method $\mathcal{A}$ to compute $\bar{\mathbf{w}}_{1-\alpha}$, and the rest training examples are only used to estimate the gradient at $\bar{\mathbf{w}}_{1-\alpha}$. In this

subsection, we propose a new approach called LastSL to address this limitation.

This approach is also under our two-phase scheme. The detailed steps are given in Algorithm 2. In phase one, we run the SGD algorithm using all the training examples, and save $\mathbf{w}_T$ in the last iteration as the approximate solution. In phase two, we apply an online-to-batch conversion procedure, similar to Algorithm 1, based on the last αT stochastic gradients in phase one to compute the final sparse solution $\widetilde{\mathbf{w}}$. α here is introduced to decide how many training examples are used in phase two.

The main difference with Algorithm 1 is that Algorithm 2 uses *all* the training examples to learn the approximate solution $\mathbf{w}_T$ while Algorithm 1 only utilizes the first $(1-\alpha)T$ training examples to learn $\bar{\mathbf{w}}_{1-\alpha}$. It would lead to a better usage of training examples. In addition, since a similar conversion procedure is applied in phase two, we have reasons to believe that our final solution $\widetilde{\mathbf{w}}$ would also be exactly sparse. These properties are confirmed by the theorem below.

Algorithm 2. Sparse Learning Based on the Last Solution

1: **Input:** strong convexity $\lambda > 0$, smoothness $L > 0$, ratio $0 < \alpha < 1$, and training examples $\{z_t = (\mathbf{x}_t, y_t)\}_{t=1}^T$,
2: Initialize $\mathbf{w}_1 = 0$
3: **for** $t = 1$ to T **do**
4: Compute the stochastic gradient $\hat{\mathbf{g}}_t = \nabla f(\mathbf{w}_t, z_t)$
5: Update

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\big(\mathbf{w}_t - \eta_t(\hat{\mathbf{g}}_t + \partial\Psi(\mathbf{w}_t)\big)$$

where $\eta_t = 1/(\lambda t)$.
6: **end for**
7: // Sparse online-to-batch conversion:
8: Compute

$$\widetilde{\mathbf{w}} = \arg\min \langle \hat{\mathbf{g}}^\alpha, \mathbf{w}\rangle + L\|\mathbf{w} - \mathbf{w}_T\|^2 + \Psi(\mathbf{w})$$

where

$$\hat{\mathbf{g}}^\alpha = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T} \nabla f(\mathbf{w}_t, z_t)$$

9: **Return:** $\widetilde{\mathbf{w}}$

Theorem 2. *Let $\delta \in (0, 1/e)$, d is the length of vector $\mathbf{g}_t$ and assume $T \geq 4$. Suppose the loss function $f(\mathbf{w}, z)$ is G-Lipschtiz continuous, λ-strongly convex and L-smooth. Then, with a probability at least $1 - 2\delta$, we have*

$$\phi(\widetilde{\mathbf{w}}_T) - \phi(\mathbf{w}_*) \leq \frac{2C_3 \log(\log(T)/\delta) G^2 L}{\lambda^2 T} + \frac{C_3 \log(\log(T)/\delta) G^2 L}{2(1-\alpha)\lambda^2 T} + \frac{4G^2 \log((d+1)/\delta)}{\alpha T L}.$$

To derive Theorem 2, we need the following Lemma 3 [26].

Lemma 3. *(Rectangular Matrix Freedman) Consider a matrix martingale $\{Y_k : k = 0, 1, 2, ...\}$ whose values are matrices with dimension $d_1 \times d_2$, and let $\{X_k : k = 1, 2, 3...\}$ be the difference sequence. Assume that the difference sequence is uniformly bounded in the sense that*

$$\|X_k\| \leq R \text{ almost surely for } k = 1, 2, 3 \ldots.$$

Define two predictable quadratic variation processes for this martingale:

$$W_{col,k} := \sum_{j=1}^{k} \mathbb{E}_{j-1}(X_j X_j^*) \text{ and}$$

$$W_{row,k} := \sum_{j=1}^{k} \mathbb{E}_{j-1}(X_j^* X_j) \text{ for } k = 1, 2, 3 \ldots$$

Then, for all $p \geq 0$ and $\sigma^2 > 0$, we have
$P\{\exists k \geq 0 : \|Y_k\| \geq p \text{ and } \max\{\|W_{col,k}\|, \|W_{row,k}\|\} \leq \sigma^2\} \leq (d_1 + d_2)\exp\{-(p^2/2)/(\sigma^2 + Rp/3)\}$

Proof for Theorem 2.

Due to the L-strongly convex of the function $h(\mathbf{w}) \triangleq \langle \hat{\mathbf{g}}^\alpha, \mathbf{w}\rangle + L\|\mathbf{w} - \mathbf{w}_T\|^2 + \Psi(\mathbf{w})$ and the definition of $\widetilde{\mathbf{w}}$, by applying Lemma 1, we have

$$\begin{aligned} &\langle \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T\rangle + L\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_T\big\|^2 + \Psi(\widetilde{\mathbf{w}}_T) \\ &\leq \langle \hat{\mathbf{g}}^\alpha, \mathbf{w}_*\rangle + L\big\|\mathbf{w}_* - \mathbf{w}_T\big\|^2 + \Psi(\mathbf{w}_*) - L\big\|\mathbf{w}_* - \widetilde{\mathbf{w}}_T\big\|^2. \end{aligned} \tag{6}$$

Analyze as we did in Theorem 1, we have:

$$\begin{aligned} &F(\widetilde{\mathbf{w}}_T) + \Psi(\widetilde{\mathbf{w}}_T) \\ &\leq F(\mathbf{w}_*) + \langle \mathbf{g}_*, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle + \frac{L}{2}\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_*\big\|^2 + \Psi(\widetilde{\mathbf{w}}_T) \\ &= F(\mathbf{w}_*) + \langle \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle + \frac{L}{2}\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_*\big\|^2 \\ &\quad + \langle \mathbf{g}_* - \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle + \Psi(\widetilde{\mathbf{w}}_T) \\ &\leq F(\mathbf{w}_*) + \langle \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle + L\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_T\big\|^2 \\ &\quad + L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 + \langle \mathbf{g}_* - \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle + \Psi(\widetilde{\mathbf{w}}_T) \\ &\overset{(6)}{\leq} F(\mathbf{w}_*) + \Psi(\mathbf{w}_*) + L\big\|\mathbf{w}_* - \mathbf{w}_T\big\|^2 - L\big\|\mathbf{w}_* - \widetilde{\mathbf{w}}_T\big\|^2 \\ &\quad + L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 + \langle \mathbf{g}_* - \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}}_T - \mathbf{w}_*\rangle \\ &\leq F(\mathbf{w}_*) + \Psi(\mathbf{w}_*) - L\big\|\mathbf{w}_* - \widetilde{\mathbf{w}}_T\big\|^2 + 2L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 \\ &\quad + \big\|\mathbf{g}_* - \hat{\mathbf{g}}^\alpha\big\|\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_*\big\| \\ &\leq F(\mathbf{w}_*) + \Psi(\mathbf{w}_*) - L\big\|\mathbf{w}_* - \widetilde{\mathbf{w}}_T\big\|^2 + 2L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 \\ &\quad + \frac{1}{4L}\big\|\mathbf{g}_* - \hat{\mathbf{g}}^\alpha\big\|^2 + L\big\|\widetilde{\mathbf{w}}_T - \mathbf{w}_*\big\|^2 \\ &= F(\mathbf{w}_*) + \Psi(\mathbf{w}_*) + 2L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 + \frac{1}{4L}\big\|\mathbf{g}_* - \hat{\mathbf{g}}^\alpha\big\|^2. \end{aligned}$$

Then, denoting $\mathbf{g}_t = \nabla F(\mathbf{w}_t)$, we have:

$$\begin{aligned} &\phi(\widetilde{\mathbf{w}}_T) - \phi(\mathbf{w}_*) \\ &\leq 2L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 + \frac{1}{2L}\Big(\Big\|\frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}(\mathbf{g}_* - \mathbf{g}_t)\Big\|^2 \\ &\quad + \Big\|\frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbf{g}_t - \hat{\mathbf{g}}_t\Big\|^2\Big) \\ &\leq 2L\big\|\mathbf{w}_T - \mathbf{w}_*\big\|^2 + \frac{L}{2\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\big\|\mathbf{w}_* - \mathbf{w}_t\big\|^2 \\ &\quad + \frac{1}{2L}\Big\|\frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbf{g}_t - \hat{\mathbf{g}}_t\Big\|^2. \end{aligned}$$

Since $\mathbf{g}_t - \hat{\mathbf{g}}_t$ is a martix martingale difference sequence with $R = 2G$, so we can use lemma 3. We can see that $\max\{\|W_{col,k}\|, \|W_{row,k}\|\} \leq 4\alpha TG^2$, for any $k \leq \alpha T$, we set $\sigma^2 = 4\alpha TG^2$ and $p = \sqrt{8G^2\alpha T \log((d+1)/\delta)}$, we have $\|\sum_{t=(1-\alpha)T+1}^{T} \mathbf{g}_t - \hat{\mathbf{g}}_t\|^2 \leq p^2 = 8G^2\alpha T \log((d+1)/\delta)$ holds with probability at least $1-\delta$.

At last, using [8, Proposition 1] to bound the first two items, we get it holds with probability at least $1-2\delta$:

$$\begin{aligned}
&\phi(\widetilde{\mathbf{w}}_T) - \phi(\mathbf{w}_*) \\
&\leq 2L\frac{C_3 \log(\log(T)/\delta)G^2}{\lambda^2 T} + \frac{L}{2}\frac{C_3 \log(\log(T)/\delta)G^2}{(1-\alpha)\lambda^2 T} \\
&\quad + \frac{1}{2L}\left(\frac{8G^2 \log((d+1)/\delta)}{\alpha T}\right) \\
&= \frac{2C_3 \log(\log(T)/\delta)G^2 L}{\lambda^2 T} + \frac{C_3 \log(\log(T)/\delta)G^2 L}{2(1-\alpha)\lambda^2 T} \\
&\quad + \frac{4G^2 \log((d+1)/\delta)}{\alpha T L}
\end{aligned}$$

□

The form of the high probability bound in Theorem 2 is similar to that in Theorem 1. From the proof above, we can see that the parameter α allows us to balance the tradeoff among the variance, prediction accuracy and the computational cost. The larger α, the lower accuracy, the smaller variance and the higher computational cost in online-to-batch conversion.

Algorithm 3. Sparse Learning Based on the Average Solution

1: **Input:** strong convexity $\lambda > 0$, smoothness $L > 0$, ratio $0 < \alpha < 1$, and training examples $\{z_t = (\mathbf{x}_t, y_t)\}_{t=1}^{T}$,
2: Initialize $\mathbf{w}_1 = 0$
3: **for** $t = 1$ to T **do**
4: Compute the stochastic gradient $\hat{\mathbf{g}}_t = \nabla f(\mathbf{w}_t, z_t)$
5: Update

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\big(\mathbf{w}_t - \eta_t(\hat{\mathbf{g}}_t + \partial\Psi(\mathbf{w}_t))\big)$$

where $\eta_t = 1/(\lambda t)$.
6: **end for**
7: // Sparse online-to-batch conversion:
8: Compute

$$\widetilde{\mathbf{w}} = \arg\min\langle \hat{\mathbf{g}}^{\alpha}, \mathbf{w}\rangle + \frac{L}{2}\|\mathbf{w} - \bar{\mathbf{w}}^{\alpha}\|^2 + \Psi(\mathbf{w})$$

where

$$\hat{\mathbf{g}}^{\alpha} = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T} \nabla f(\mathbf{w}_t, z_t),$$

$$\text{and } \bar{\mathbf{w}}^{\alpha} = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T} \mathbf{w}_t$$

9: **Return:** $\widetilde{\mathbf{w}}$

Nevertheless, there is a potential drawback in Algorithm 2 as well: there is a factor λ^{-2} in the bound of Theorem 2, while the corresponding factor in Theorem 1 is λ^{-1}. It seems that when the parameter λ is small, this factor would make Algorithm 2 worse than Algorithm 1. This phenomenon will be discussed in detail in the next section.

4.3 Sparse Learning Based on the Average Solution

Firstly, we would like to give some discussion about the limitations in Algorithm 2 in the perspective of the algorithm self and the theoretical bounds. Then we will design a new approach called AverageSL to deal with these weaknesses.

The limitations of Algorithm 2 are:

- The final solution depends heavily on the last solution $\mathbf{w}_T$, which leads to a relatively large variance.
- The factor λ^{-2} in the bound of Theorem 2 is significantly worse than the factor λ^{-1} in Theorem 1.

Aiming at the shortages above, we design a new algorithm, whose the detail steps are described in Algorithm 3.

Algorithm 3 is also under our two-phase scheme and very similar to Algorithm 2. Unlike Algorithm 2, we average and return a suffix as the approximate solution instead of returning the last solution $\mathbf{w}_T$, namely

$$\bar{\mathbf{w}}^{\alpha} = (\mathbf{w}_{(1-\alpha)T+1} + \mathbf{w}_{(1-\alpha)T+2} + \cdots + \mathbf{w}_T)/(\alpha T)$$

for some constant $\alpha \in (0,1)$, which plays a similar role as the corresponding parameter in Algorithm 2. Then, we apply the online-to-batch conversion procedure based on the average gradient $\hat{\mathbf{g}}^{\alpha}$ to $\bar{\mathbf{w}}^{\alpha}$ to reach an exactly sparse solution $\widetilde{\mathbf{w}}$.

We present the convergence performance of this algorithm in the theorem below:

Theorem 3. *Let $\delta \in (0, 1/e)$, d be the length of vector $\mathbf{w}_*$ and assume $T \geq 4$. Suppose $F(\cdot)$ is λ-strongly convex, L -smooth and let M be a smooth parameter satisfies that $\|\nabla^2[\nabla F(\mathbf{w})]_i\|_2 \leq M$, $\|\cdot\|_2$ here is the spectral norm. In addition, we assume ϕ's stochastic gradient at $\mathbf{w}_t$ denoted as $\partial\hat{\phi}_t$ satisfies that $\|\partial\hat{\phi}_t\|^2 \leq \tilde{G}^2$. Then it holds with probability at least $1-2\delta$ that for any $t \leq T$,*

$$\begin{aligned}
&\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \\
&\leq \frac{C_4 \log(\log(T)/\delta)\tilde{G}^2}{\lambda\alpha T} + \frac{C_5 \log(\log(T)/\delta)\tilde{G}^2}{\lambda(1-\alpha)T} \\
&\quad + \frac{C_6 \log(1/(1-\alpha))\tilde{G}^2}{\alpha\lambda T} + \frac{8G^2 \log((d+1)/\delta)}{\alpha L T} \\
&\quad + \frac{C_3^2 (\log(\log(T)/\delta))^2 G^4 d M^2}{\lambda^4(1-\alpha)^2\alpha^2 L T^2}.
\end{aligned}$$

To derive Theorem 3, we need the lemma bellow, which is an extension of Lemma 2 in [8].

Lemma 4. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$. Suppose ϕ is $\lambda-$strongly convex over a convex set $\mathcal{W}$, and its stochastic gradient $\partial\hat{\phi}_t = \nabla f(\mathbf{w}_t, z_t) + \partial\Psi(\mathbf{w}_t)$ satisfies that $\|\partial\hat{\phi}_t\|^2 \leq \tilde{G}^2$. Then if we pick $\eta_t = 1/\lambda t$, it holds with probability at least $1-\delta$, that for any $t \leq T$,*

$$\begin{aligned}
\phi(\bar{\mathbf{w}}^{\alpha}) - \phi(\mathbf{w}_*) &\leq \frac{C_4 \log(\log(T)/\delta)\tilde{G}^2}{\lambda\alpha T} \\
&\quad + \frac{C_5 \log(\log(T)/\delta)\tilde{G}^2}{\lambda(1-\alpha)T} + \frac{C_6 \log(1/(1-\alpha))\tilde{G}^2}{\alpha\lambda T},
\end{aligned}$$

where C_4, C_5 and C_6 are three constants.

Proof for Lemma 4.

Here, we use $\partial\phi_t$ to be the gradient of $\phi(\mathbf{w}_t)$. Since

$$\begin{aligned}
\|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2 \leq &\|\mathbf{w}_t - \mathbf{w}_*\|^2 - 2\eta_t\langle\partial\phi_t, \mathbf{w}_t - \mathbf{w}_*\rangle \\
&+ \eta_t^2\tilde{G}^2 + 2\eta_t\langle\partial\phi_t - \partial\hat{\phi}_t, \mathbf{w}_t - \mathbf{w}_*\rangle.
\end{aligned}$$

We have

$$\langle \partial\phi_t, \mathbf{w}_t - \mathbf{w}_* \rangle \leq \frac{\eta_t \tilde{G}^2}{2} + \langle \partial\phi_t - \partial\hat{\phi}_t, \mathbf{w}_t - \mathbf{w}_* \rangle + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta_t}.$$

Summing over $t = (1-\alpha)T+1, \ldots, T$,we get:

$$\begin{aligned}
&\sum_{t=(1-\alpha)T+1}^{T} \langle \partial\phi_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
&\leq \sum_{t=(1-\alpha)T+1}^{T} \frac{\eta_t \tilde{G}^2}{2} + \sum_{t=(1-\alpha)T+1}^{T} \langle \partial\phi_t - \partial\hat{\phi}_t, \mathbf{w}_t - \mathbf{w}_* \rangle \qquad (7) \\
&\quad + \sum_{t=(1-\alpha)T+1}^{T} \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta_t}.
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\phi(\bar{\mathbf{w}}_T^\alpha) - \phi(\mathbf{w}_*) \\
&\leq \frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \big(\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\big) \\
&\leq \frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \langle \partial\phi_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
&\overset{(7)}{\leq} \frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \left(\frac{\eta_t \tilde{G}^2}{2} + \langle \partial\phi_t - \partial\hat{\phi}_t, \mathbf{w}_t - \mathbf{w}_* \rangle \right. \\
&\quad \left. + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta_t} \right) \\
&\leq \frac{1}{2\alpha T} \left(\frac{1}{\eta_{(1-\alpha)T+1}} \|\mathbf{w}_{(1-\alpha)T+1} - \mathbf{w}_*\|^2 \right. \\
&\quad + \sum_{t=(1-\alpha)T+1}^{T} \|\mathbf{w}_t - \mathbf{w}_*\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \\
&\quad \left. + 2 \sum_{t=(1-\alpha)T+1}^{T} \frac{\tilde{G}^2 \eta_t}{2} + \langle \partial\phi_t - \partial\hat{\phi}_t, \mathbf{w}_t - \mathbf{w}_* \rangle \right).
\end{aligned}$$

To bound the last term, we consider $Z_i = \langle \mathbf{g}_i - \partial\hat{\phi}_i, \mathbf{w}_i - \mathbf{w}_* \rangle . \|\mathbf{g}_i - \partial\hat{\phi}_i\| \leq 2\tilde{G}$, so $\mathrm{Var}_{i-1}(Z_i) \leq 4\tilde{G}^2 \|\mathbf{w}_i - \mathbf{w}_*\|^2$. Also, we have $\|\mathbf{w}_i - \mathbf{w}_*\| \leq 2\tilde{G}/\lambda$, thus $|Z_i| \leq 4\tilde{G}^2/\lambda$. Using [8, Lemma 3] , we have, as long as $T \geq 4$ and $\delta \in (0, 1/e)$, then with probability at least $1-\delta$, for all $t \leq T$, it holds that

$$\sum_{i=(1-\alpha)T+1}^{T} Z_i \leq 8\tilde{G}\Delta \sqrt{\log\left(\frac{\log(T)}{\delta}\right)}, \quad \text{where}$$

$$\Delta = \max\left\{ \sqrt{\sum_{i=(1-\alpha)T+1}^{t} \|\mathbf{w}_i - \mathbf{w}_*\|^2}, \frac{\tilde{G}}{\lambda} \sqrt{\log\left(\frac{\log(T)}{\delta}\right)} \right\},$$

Just by using [8, Proposition 1] to bound the first two terms, we can immediately get: it holds with probability at least $1-\delta$ that for any $t \leq T$

$$\|\mathbf{w}_t - \mathbf{w}_*\|^2 \leq \big(624 \log(\log(T)/\delta) + 1\big)\tilde{G}^2/(\lambda^2 t).$$

In addition, since $\sum_{t=(1-\alpha)T+1}^{T} \eta_t \leq \log(1/(1-\alpha))/\lambda$, and by combining all the analysis above, we can get:

$$\begin{aligned}
&\phi(\bar{\mathbf{w}}_T^\alpha) - \phi(\mathbf{w}_*) \\
&\leq \frac{1}{2\alpha T} \Big(\frac{(624 \log(\log(T)/\delta) + 1)\tilde{G}^2}{\lambda} + \frac{\log(1/(1-\alpha))}{\lambda} \\
&\quad + \alpha T \frac{(624 \log(\log(T)/\delta) + 1)\tilde{G}^2}{\lambda((1-\alpha)T+1)} \\
&\quad + 4G\Delta \sqrt{\log(\log(T)/\delta)} \Big) \\
&= \frac{C_4 \log(\log(T)/\delta)\tilde{G}^2}{\lambda\alpha T} + \frac{C_5 \log(\log(T)/\delta)\tilde{G}^2}{\lambda(1-\alpha)T} \\
&\quad + \frac{C_6 \log(1/(1-\alpha))\tilde{G}^2}{\alpha\lambda T}.
\end{aligned}$$

□

Proof for Theorem 3.

First, from the definition of $\widetilde{\mathbf{w}}$ and Lemma 1, we have

$$\begin{aligned}
&\langle \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}} \rangle + \frac{L}{2} \|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha\|^2 + \Psi(\widetilde{\mathbf{w}}) \\
&\leq \langle \hat{\mathbf{g}}^\alpha, \bar{\mathbf{w}}^\alpha \rangle - \frac{L}{2} \|\bar{\mathbf{w}}^\alpha - \widetilde{\mathbf{w}}\|^2 + \Psi(\bar{\mathbf{w}}^\alpha).
\end{aligned} \qquad (8)$$

So, we can get

$$\begin{aligned}
&F(\widetilde{\mathbf{w}}) + \Psi(\widetilde{\mathbf{w}}) \\
&\leq F(\bar{\mathbf{w}}^\alpha) + \langle \nabla F(\bar{\mathbf{w}}^\alpha), \widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha \rangle + \frac{L}{2} \|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha\|^2 + \Psi(\widetilde{\mathbf{w}}) \\
&= F(\bar{\mathbf{w}}^\alpha) + \langle \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha \rangle + \frac{L}{2} \|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha\|^2 + \Psi(\widetilde{\mathbf{w}}) \\
&\quad + \langle \nabla F(\bar{\mathbf{w}}^\alpha) - \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha \rangle \\
&\overset{(8)}{\leq} F(\bar{\mathbf{w}}^\alpha) + \Psi(\bar{\mathbf{w}}^\alpha) - \frac{L}{2} \|\bar{\mathbf{w}}^\alpha - \widetilde{\mathbf{w}}\|^2 \\
&\quad + \langle \nabla F(\bar{\mathbf{w}}^\alpha) - \hat{\mathbf{g}}^\alpha, \widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha \rangle \\
&\leq F(\bar{\mathbf{w}}^\alpha) + \Psi(\bar{\mathbf{w}}^\alpha) - \frac{L}{2} \|\bar{\mathbf{w}}^\alpha - \widetilde{\mathbf{w}}\|^2 \\
&\quad + \frac{1}{2L} \|\nabla F(\bar{\mathbf{w}}^\alpha) - \hat{\mathbf{g}}^\alpha\|^2 + \frac{L}{2} \|\widetilde{\mathbf{w}} - \bar{\mathbf{w}}^\alpha\|^2 \\
&\leq F(\bar{\mathbf{w}}^\alpha) + \Psi(\bar{\mathbf{w}}^\alpha) + \frac{1}{2L} \|\nabla F(\bar{\mathbf{w}}^\alpha) - \hat{\mathbf{g}}^\alpha\|^2.
\end{aligned} \qquad (9)$$

Thus, we have:

$$\begin{aligned}
&\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \\
&\leq \phi(\bar{\mathbf{w}}^\alpha) - \phi(\mathbf{w}_*) + \frac{1}{2L} \|\nabla F(\bar{\mathbf{w}}^\alpha) - \hat{\mathbf{g}}^\alpha\|^2 \\
&\leq \phi(\bar{\mathbf{w}}^\alpha) - \phi(\mathbf{w}_*) + \frac{1}{\alpha^2 T^2 L} \Big(\Big\| \sum_{t=(1-\alpha)T+1}^{T} \mathbf{g}_t - \hat{\mathbf{g}}_t \Big\|^2 \qquad (10) \\
&\quad + \Big\| \sum_{t=(1-\alpha)T+1}^{T} \big(\nabla F(\bar{\mathbf{w}}^\alpha) - \nabla F(\mathbf{w}_t)\big) \Big\|^2 \Big).
\end{aligned}$$

Then, we try to bound the item $\Big\| \sum_{t=(1-\alpha)T+1}^{T} \big(\nabla F(\bar{\mathbf{w}}^\alpha) - \nabla F(\mathbf{w}_t)\big) \Big\|^2$ below. For simplicity, we note $f_i(\mathbf{w}) \triangleq [\nabla F(\mathbf{w})]_i$ from now on and thus

$$\begin{aligned}
&\Big\| \sum_{t=(1-\alpha)T+1}^{T} \big(\nabla F(\bar{\mathbf{w}}^\alpha) - \mathbf{g}_t\big) \Big\|^2 \\
&= \sum_{i=1}^{d} \left(\sum_{t=(1-\alpha)T+1}^{T} \big(f_i(\bar{\mathbf{w}}^\alpha) - f_i(\mathbf{w}_t)\big) \right)^2.
\end{aligned}$$

Without loss of generality, we consider the component

$$\delta_i \triangleq \left(\sum_{t=(1-\alpha)T+1}^{T} \big(f_i(\bar{\mathbf{w}}^\alpha) - f_i(\mathbf{w}_t)\big) \right)^2. \tag{11}$$

We conduct Taylor Expansion on $f_i(\bar{\mathbf{w}}^\alpha)$ and $f_i(\mathbf{w}_t)$ at the optimal solution $\mathbf{w}_*$, and note $\boldsymbol{\epsilon}_t = \mathbf{w}_t - \mathbf{w}_*$ and $\bar{\boldsymbol{\epsilon}} = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T} \boldsymbol{\epsilon}_t = \bar{\mathbf{w}}^\alpha - \mathbf{w}_*$. We can get:

$$\begin{aligned} f_i(\bar{\mathbf{w}}^\alpha) &= f_i(\mathbf{w}_* + \bar{\boldsymbol{\epsilon}}) = f_i(\mathbf{w}_*) + \nabla f_i(\mathbf{w}_*)^T \bar{\boldsymbol{\epsilon}} \\ &\quad + \frac{1}{2}\bar{\boldsymbol{\epsilon}}^T \nabla^2 f_i(\mathbf{w}_* + \bar{\theta}\bar{\boldsymbol{\epsilon}})\bar{\boldsymbol{\epsilon}} \text{ for some } \bar{\theta} \in (0,1) \end{aligned} \tag{12}$$

$$\begin{aligned} f_i(\mathbf{w}_t) &= f_i(\mathbf{w}_* + \boldsymbol{\epsilon}_t) = f_i(\mathbf{w}_*) + \nabla f_i(\mathbf{w}_*)^T \boldsymbol{\epsilon}_t \\ &\quad + \frac{1}{2}\boldsymbol{\epsilon}_t^T \nabla^2 f_i(\mathbf{w}_* + \theta_t \boldsymbol{\epsilon}_t)\boldsymbol{\epsilon}_t \text{ for some } \theta_t \in (0,1). \end{aligned} \tag{13}$$

Plug equations (12) and (13) into (11), we can get:

$$\begin{aligned} \delta_i &= \left(\frac{\alpha T}{2}\bar{\boldsymbol{\epsilon}}^T \nabla^2 f_i(\mathbf{w}_* + \bar{\theta}\bar{\boldsymbol{\epsilon}})\bar{\boldsymbol{\epsilon}} - \frac{1}{2}\sum_{t=(1-\alpha)T+1}^{T} \boldsymbol{\epsilon}_t^T \nabla^2 f_i(\mathbf{w}_* + \theta_t \boldsymbol{\epsilon}_t)\boldsymbol{\epsilon}_t \right)^2 \\ &\le \left(\frac{\alpha T M}{2}\big\|\bar{\mathbf{w}}^\alpha - \mathbf{w}_*\big\|^2 + \frac{M}{2}\sum_{t=(1-\alpha)T+1}^{T} \big\|\mathbf{w}_t - \mathbf{w}_*\big\|^2 \right)^2. \end{aligned} \tag{14}$$

Plugging the last inequation into (10), we have:

$$\begin{aligned} &\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \\ &\le \phi(\bar{\mathbf{w}}^\alpha) - \phi(\mathbf{w}_*) + \frac{1}{\alpha^2 T^2 L}\Big\| \sum_{t=(1-\alpha)T+1}^{T} \mathbf{g}_t - \hat{\mathbf{g}}_t \Big\|^2 \\ &\quad + \frac{dM^2}{4\alpha^2 L T^2}\left\{ \alpha T \big\|\bar{\mathbf{w}}^\alpha - \mathbf{w}_*\big\|^2 + \sum_{t=(1-\alpha)T+1}^{T} \big\|\mathbf{w}_t - \mathbf{w}_*\big\|^2 \right\}^2. \end{aligned}$$

Thus, we can use [8, Lemma 1 and Proposition 1] to bound the items $\big\|\bar{\mathbf{w}}^\alpha - \mathbf{w}_*\big\|^2$ and $\big\|\mathbf{w}_t - \mathbf{w}_*\big\|^2$.

Finally, by using Lemma 4 to bound $\phi(\bar{\mathbf{w}}^\alpha) - \phi(\mathbf{w}_*)$, and analyse the second item $\big\|\sum_{t=(1-\alpha)T+1}^{T} \mathbf{g}_t - \hat{\mathbf{g}}_t\big\|^2$ by using Lemma 3 as we did in Theorem 2, we can get, it holds with probability at least $1 - 3\delta$ that,

$$\begin{aligned} &\phi(\widetilde{\mathbf{w}}) - \phi(\mathbf{w}_*) \\ &\le \frac{C_4 \log(\log(T)/\delta)\tilde{G}^2}{\lambda\alpha T} + \frac{C_5 \log(\log(T)/\delta)\tilde{G}^2}{\lambda(1-\alpha)T} \\ &\quad + \frac{C_6 \log(1/(1-\alpha))\tilde{G}^2}{\alpha\lambda T} + \frac{8G^2 \log((d+1)/\delta)}{\alpha L T} \\ &\quad + \frac{C_3^2 \left(\log(\log(T)/\delta)\right)^2 G^4 d M^2}{\lambda^4 (1-\alpha)^2 \alpha^2 L T^2} \end{aligned}$$

□

In Theorem 3, we assume the stochastic gradient of ϕ is bounded by $\tilde{G}$. This assumption is widely used in the exiting work such as [8] and it is acceptable since the magnitude of the regularizer $\Psi(\mathbf{w})$ is always small.

Theorem 3 implies that we can improve one item in the high probability bound of our Algorithm 2 from $\mathcal{O}(\log(\log(T)/\delta)/(\lambda^2 T))$ to $\mathcal{O}(\log(\log(T)/\delta)/(\lambda T)) + \mathcal{O}((\log(\log(T)/\delta)^2)/(\lambda^4 T^2)) \approx \mathcal{O}(\log(\log(T)/\delta)/(\lambda T))$. It would be much better than the bound in Theorem 2 when the iteration number T is large enough. This success is benefited from the fact that $\hat{\mathbf{g}}^\alpha$ is a better approximation for $\partial F(\bar{\mathbf{w}}^\alpha)$ than for $\partial F(\mathbf{w}_T)$. Precisely, $\hat{\mathbf{g}}^\alpha$ is a second-order approximation for $\partial F(\bar{\mathbf{w}}^\alpha)$, which can be seen from (14), while only a first-order approximation for $\partial F(\mathbf{w}_T)$. So the online-to-batch conversion in Algorithm 3 in much more accurate.

At last, we point out that our methods are much better than the recent methods like ORDA at the high probability bound, which is demonstrated in Table 1.

TABLE 1
Summary of the High Probability Bounds

Methods	High Probability Bounds
ORDA	$\mathcal{O}(\sqrt{\log(1/\delta)/T})$ or $\mathcal{O}(1/(\delta\lambda T))$
OptimalSL	$\mathcal{O}\Big(\frac{\log(\log(T)/\delta)}{\lambda T}\Big) + \mathcal{O}\Big(\frac{(\log(1/\delta))^2}{T}\Big)$
LastSL	$\mathcal{O}\Big(\frac{\log(\log(T)/\delta)}{\lambda^2 T}\Big) + \mathcal{O}\Big(\frac{\log(1/\delta)}{T}\Big)$
AverageSL	$\mathcal{O}\Big(\frac{\log(\log(T)/\delta)}{\lambda T}\Big) + \mathcal{O}\Big(\frac{\log(1/\delta)}{T}\Big) + \mathcal{O}\Big(\frac{(\log(\log(T)/\delta))^2}{\lambda^4 T^2}\Big)$

5 Experiments

Having demonstrated the convergence performances of the proposed approaches, we now turn to the empirical study.

As we mentioned above, many methods have been proposed in the optimization and machine learning literatures for minimizing Eq.(1). The baseline algorithms for comparison are chosen with the following considerations: they should cover the recent well-known works on SO, SCO, online sparse learning and etc. At last, we choose the following three well-known algorithms as baselines.

- ORDA [2]: a SCO algorithm that yields the optimal expected convergence rate $\mathcal{O}(1/\lambda T)$ and high probability bound $\mathcal{O}(\sqrt{\log(1/\delta)/T})$. And it attaches great importance to the sparsity of the final solution.
- α-SGD [8]: a simple but effective SO algorithm with the expected convergence rate $\mathcal{O}(1/\lambda T)$ and high probability bound $\mathcal{O}(\log(\log(T)/\delta)/\lambda T)$.
- FOBOS [23]: an online sparse learning algorithm focuses on the sparsity of the solution.

We take α-SGD as the algorithm $\mathcal{A}$ in OptimalSL and divide our experiment section into two parts: experiments on the synthesized datasets and on the real-word datasets. We evaluate the performance of our approaches on two aspects: (i) whether the learned $\widetilde{\mathbf{w}}$ is close to the optimal solution, (ii) whether the learned $\widetilde{\mathbf{w}}$ is sparse and recovers most of the relevant features.

5.1 Experiments on the Synthesized Datasets

5.1.1 Experimental Model

Similar to work [2], [27], we solve the optimization problem, whose object function is comprised of three items: least square item, ℓ_2-norm regularizer and ℓ_1-norm regularizer, i.e., $\min_{\mathbf{w}\in\mathbb{R}^d} \phi(\mathbf{w}) = f(\mathbf{w}) + h(\mathbf{w}), where\ f(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{\mathbf{a},b}((\langle \mathbf{w}, \mathbf{a}\rangle - b)^2) + \frac{\rho}{2}\|\mathbf{w}\|_2^2$ and $h(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$. $\mathbf{a}$ is a random vector valued in $\mathbb{R}^d$. The response b is generated by $b = \langle \mathbf{a}, \mathbf{w}^*\rangle + \epsilon$, where ϵ is a random noise and $\mathbf{w}^*$ is a constant vector in $\mathbb{R}^d$.

TABLE 2
Numerical Results on ℓ_1 Regularized Linear Regression Problem with d=100, $\lambda = 0.1, \rho = 0.1$

$\sigma_e^2 = 1$	$d = 100, N = 200000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	5.7	0.99	0.99	0.67	3.2e-08	14.8
α-SGD	5.7	1	0.99	0.67	3.3e-08	13.5
ORDA	5.7	0.99	0.5	0.67	1.2e-07	20.8
LastSL	**5.7**	**0.5**	**0.5**	**1**	**7.8e-08**	**14.2**
OptimalSL	**5.7**	**0.5**	**0.5**	**1**	**4.5e-08**	**13.3**
AverageSL	**5.7**	**0.5**	**0.5**	**1**	**3.0e-08**	**14.2**
$\sigma_e^2 = 4$	$d = 100, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	7.2	1	1	0.67	1.1e-07	29.5
α-SGD	7.2	1	0.99	0.67	2.7e-08	27.2
ORDA	7.2	1	0.5	0.67	1.9e-07	41.9
LastSL	**7.2**	**0.5**	**0.5**	**1**	**4.7e-08**	**28.3**
OptimalSL	**7.2**	**0.5**	**0.5**	**1**	**2.2e-08**	**26.5**
AverageSL	**7.2**	**0.5**	**0.5**	**1**	**3.3e-08**	**28.3**
$\sigma_e^2 = 25$	$d = 100, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	17.7	1	1	0.67	7.4e-07	29.5
α-SGD	17.7	1	1	0.67	3.0e-07	27.1
ORDA	17.7	1	0.5	0.67	1.6e-06	41.8
LastSL	**17.7**	**0.5**	**0.5**	**1**	**7.2e-07**	**28.3**
OptimalSL	**17.7**	**0.5**	**0.5**	**1**	**2.2e-07**	**26.4**
AverageSL	**17.7**	**0.5**	**0.5**	**1**	**2.4e-07**	**28.2**
$\sigma_e^2 = 100$	$d = 100, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	55.2	1	1	0.67	1.6e-05	29.6
α-SGD	55.2	1	1	0.67	4.3e-06	27.2
ORDA	55.2	1	0.74	0.67	4.8e-05	41.7
LastSL	**55.2**	**0.57**	**0.57**	**0.93**	**4.9e-06**	**28.3**
OptimalSL	**55.2**	**0.5**	**0.5**	**1**	**4.8e-06**	**26.4**
AverageSL	**55.2**	**0.5**	**0.5**	**1**	**4.5e-06**	**28.3**

TABLE 3
Numerical Results on ℓ_1 Regularized Linear Regression Problem with d=1000, $\lambda = 0.1, \rho = 0.1$

$\sigma_e^2 = 1$	$d = 1000, N = 400,000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	52.5	1	1	0.67	2.4e-05	41.1
α-SGD	52.5	1	1	0.67	7.0e-06	36.4
ORDA	52.5	1	0.54	0.67	2.5e-05	59.4
LastSL	**52.5**	**0.5**	**0.5**	**1**	**5.5e-06**	**37.5**
OptimalSL	**52.5**	**0.5**	**0.5**	**1**	**5.3e-06**	**34.9**
AverageSL	**52.5**	**0.5**	**0.5**	**1**	**7.4e-06**	**37.7**
$\sigma_e^2 = 4$	$d = 1000, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	54.1	1	1	0.67	1.9e-05	41
α-SGD	54	1	1	0.67	5.3e-06	36.1
ORDA	54	1	0.55	0.67	4.9e-05	59.1
LastSL	**54**	**0.5**	**0.5**	**1**	**5.6e-06**	**37.2**
OptimalSL	**54**	**0.5**	**0.5**	**1**	**6.6e-06**	**34.7**
AverageSL	**54**	**0.5**	**0.5**	**1**	**6.4e-06**	**37.4**
$\sigma_e^2 = 25$	$d = 1000, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	64.6	1	1	0.67	9.7e-05	41.3
α-SGD	64.5	1	1	0.67	1.4e-05	36.6
ORDA	64.6	1	0.62	0.67	1.3e-04	59.6
LastSL	**64.5**	**0.52**	**0.52**	**0.98**	**2.1e-05**	**37.6**
OptimalSL	**64.5**	**0.5**	**0.5**	**1**	**9.8e-06**	**35.1**
AverageSL	**64.5**	**0.5**	**0.5**	**1**	**1.1e-05**	**37.8**
$\sigma_e^2 = 100$	$d = 1000, N = 400000$					
	Obj	ED	TD	SSR	Var	time
FOBOS	102.4	1	1	0.67	1.8e-04	41.1
α-SGD	102.1	1	1	0.67	9.8e-05	36.3
ORDA	102.3	1	0.84	0.67	6.4e-04	59.5
LastSL	**102.2**	**0.63**	**0.63**	**0.88**	**1.8e-04**	**37.5**
OptimalSL	**102.1**	**0.5**	**0.5**	**1**	**6.7e-05**	**34.9**
AverageSL	**102**	**0.5**	**0.5**	**1**	**4.6e-05**	**37.6**

The parameters ρ and λ are two constants used to balance the bias and the simplicity of the model.

5.1.2 Parameter Setting

To describe the parameters clearly, we separate them into three parts, i.e., for model $\{\lambda, \rho, \sigma_e\}$, for training dataset $\{\mathbf{a}, b, \mathbf{w}_*\}$ and for algorithms $\{\alpha\}$. First, for the model, we set the weights $\lambda = 0.1, \rho = 0.1$, and vary σ_e in the range of $[1, 2, 3, ..., 10]$. Secondly, for the training dataset, we let the entries of $\mathbf{a}$ be independent identically distributed variables which are sampled from the same uniform distribution $U(-1, 1)$. The noise ϵ of the response b is $N(0, \sigma_e^2)$ distributed, and the vector $\mathbf{w}^*$ satisfies that $[\mathbf{w}^*]_i = 1$ for $1 \leq i \leq \frac{d}{2}$ and 0 otherwise. We set $d = 100$ and $1,000$ to evaluate our methods both when the dimension is low and relatively high. In addition, we sample the random variable pair $(\mathbf{a}, b)$ from this model 200,000 times when $\sigma_e = 1, d = 100$ and 400,000 times otherwise, so the training dataset size N in our experiment is $200,000$ or $400,000$. At last, we set the $\alpha = 0.3$ for α-SGD and our methods.

5.1.3 The Optimal Solution of Our Model

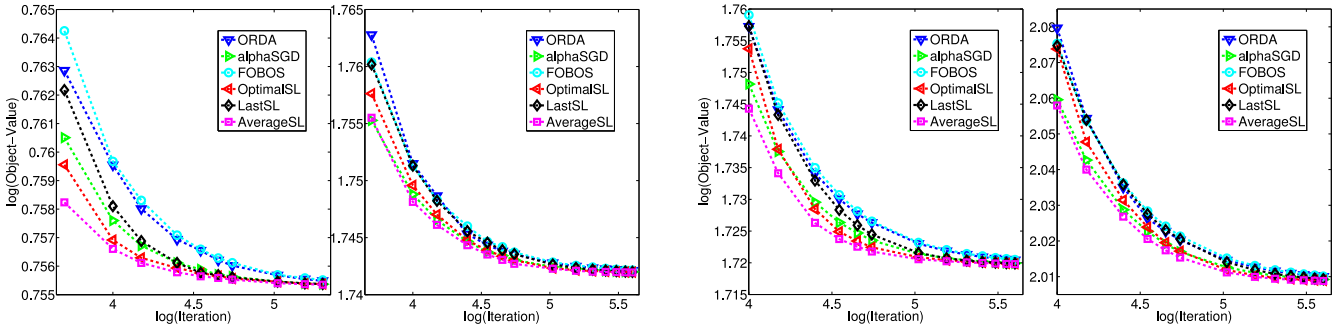
It is easy to verify that under our assumptions above, we have $\frac{1}{2}\mathbb{E}_{\mathbf{a},b}((\mathbf{a}^T\mathbf{w} - b)^2) = \frac{1}{6}\|\mathbf{w} - \mathbf{w}^*\|_2^2 + \frac{1}{2}\sigma_e^2$. Fortunately, we can calculate both the exact objective function value and the exact optimal solution $\mathbf{w}_*$ in this special case, i.e., $\phi(\mathbf{w}) = \frac{1}{6}\|\mathbf{w} - \mathbf{w}^*\|_2^2 + \frac{1}{2}\sigma_e^2 + \frac{\rho}{2}\|\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1$ and $[\mathbf{w}_*]_i = \frac{7}{13}$ *for* $i \leq d/2$ and 0, otherwise.

5.1.4 Evaluation Metrics

Similar to [13], we evaluate the performances of our algorithms mainly from the following three aspects:

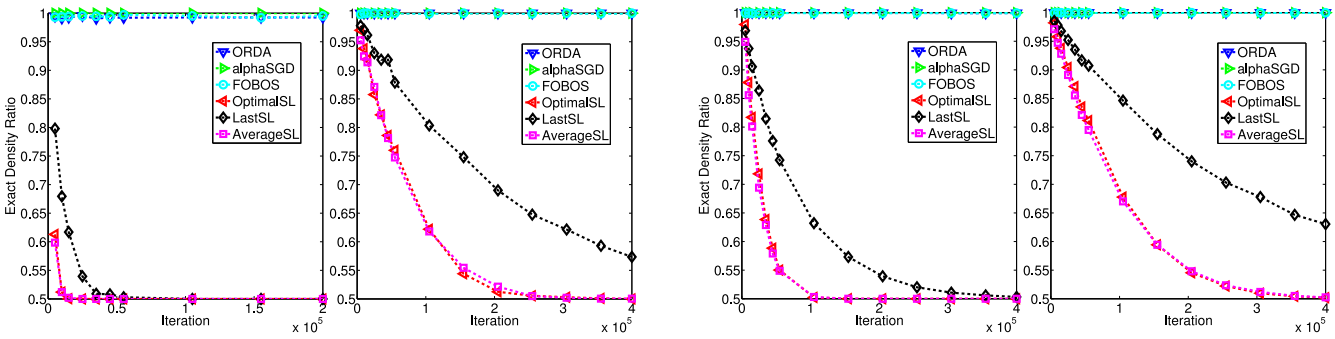
1) the efficiency of our algorithms.
2) the sparsity of the solutions over iterations.
3) the recovery of the support set of $\mathbf{w}_*$.

Specifically, for the efficiency, we calculate the objective values of all the algorithms over iterations and give the running time in Tables 2 and 3. For the sparsity, we use the following two metrics: the exact density ratio (**ED** for short), which is computed as $\frac{1}{d}\sum_{i=1}^{d} I([\mathbf{w}]_i \neq 0)$, and the truncated sparse ratio (**TD** for short) computed as $\frac{1}{d}\sum_{i=1}^{d} I(|[\mathbf{w}]_i| > \epsilon_r)$, where ϵ_r is set to be 10^{-6}. For the recovery of the support set of $\mathbf{w}_*$, we use the widely used metric $\mathbf{SSR}(\mathbf{w})$. It is defined as $\mathbf{SSR}(\mathbf{w}) = 2|\mathcal{S}(\mathbf{w}) \cap \mathcal{S}(\mathbf{w}_*)|/(|\mathcal{S}(\mathbf{w})| + |\mathcal{S}(\mathbf{w}_*)|)$, where $\mathcal{S}(\mathbf{w})$ is the support set of $\mathbf{w}$ composed of the nonzero components of $\mathbf{w}$, $|\mathcal{S}(\mathbf{w})|$ means the cardinality of $\mathcal{S}(\mathbf{w})$. We run each experiment 100 times and report the averaged results.



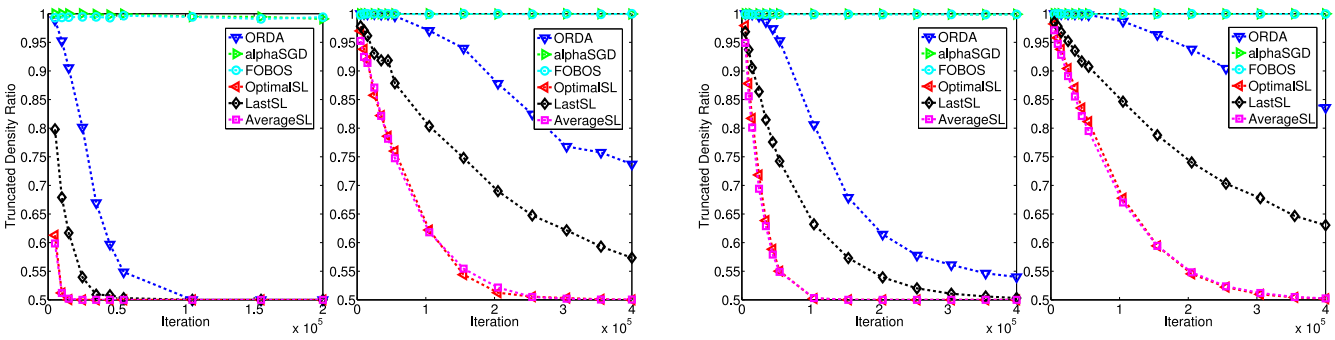
(a) σ_e=1 (left) and 10 (right) (b) σ_e=1 (left) and 10 (right)

Fig. 2. Objective values over iterations with parameter ρ=0.1, λ=0.1 and d=100 (a) and d=1,000 (b).



(a) σ_e=1 (left) and 10 (right) (b) σ_e=1 (left) and 10 (right)

Fig. 3. Exact density ratio over iterations with parameter ρ=0.1, λ=0.1 and d=100 (a) and d=1,000 (b).



(a) σ_e=1 (left) and 10 (right) (b) σ_e=1 (left) and 10 (right)

Fig. 4. Truncated density ratio over iterations with parameter ρ=0.1, λ=0.1 and d=100 (a) and d=1000 (b).

5.1.5 *Experimental Results*

Our experiment results are mainly comprised of two parts: the quality of the solutions with different iteration times and the numerical results with the maximal iteration times.

For the first part, we show the objective function's value, ED, TD and SSR of the solutions obtained by these algorithms under different noise level σ_e and different dimensions after different iteration times in Figs. 2, 3, 4, and 5.

From Fig. 2, we observe that the three proposed algorithms are comparable to, if not better than, the baselines in reducing the value of the objective function over iterations.

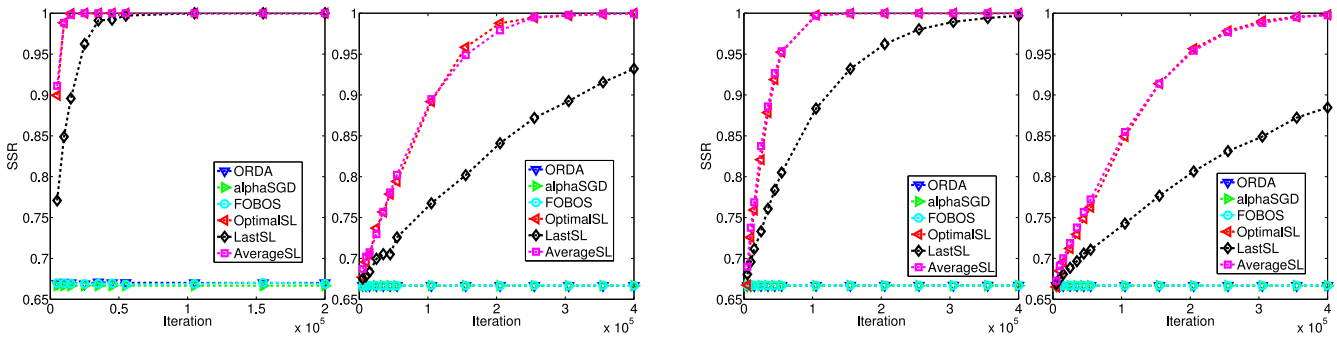
In Figs. 3 and 4, the curves of our methods decrease rapidly, while the compared algorithms decrease much more slowly or even do not decrease at all. So our algorithms outperform the baselines at sparse recovery ability.

The SSR curves in Fig. 5 show that our methods can not only make the solutions more and more sparse over iterations, but also recover most of the elements in $\mathcal{S}(\mathbf{w}_*)$.

More importantly, the results in Fig. 4 show that our methods are much more robust to noise than others. The difference between the figures (a) and (b) also reflects that our methods are more insensitive to data dimension.

For the second part, we summarizes the evaluation results of different algorithms with the maximal iteration times in Tables 2 and 3. They show that besides yielding comparable value for the objective function, the solutions found by our algorithms are significantly sparser than the ones found by the baseline algorithms. The variances of our methods OptimalSL and AverageSL are much smaller than ORDA and FOBOS, which is consistent with our high probability bounds in the last section.

Tables 2 and 3 also demonstrate that our methods are more powerful when dealing with the data with high noise level. The baseline algorithms can not learn sparse solutions when the data is noisy, especially in high dimensional problems, while our methods still work well.



(a) σ_e=1 (left) and 10 (right) (b) σ_e=1 (left) and 10 (right)

Fig. 5. SSR over iterations with parameter ρ=0.1, λ=0.1 and d=100 (a) and d=1000 (b).

As we discussed in the subsection "Sparse Learning based on the Last Solution" that our approach OptimalSL can not make use of the training examples to the fullest, this phenomenon will be more obvious in high dimensional problems. To confirm this point, we conduct an extra experiment to compare the performances of our three approaches when $d = 1,000, N = 10,000$. Other parameters are the same with the experiments above. We show the results in Fig. 6. It is obvious that the objective value of OptimalSL decreases more slowly than AverageSL and LastSL, which is consistent with our analysis.

In a word, our methods are superior to other methods at the aspect of sparse recover ability and they are much more robust to the noise and dimension. In addition, our three methods have similar performances and it is hard to say which is the best, especially for OptimalSL and AverageSL. It may results from the fact that our model here is relatively simple, so our methods are all adequate for this task.

5.2 Experiments on Real-World Dataset

5.2.1 Dataset

To further demonstrate the effectiveness of our methods, we conduct binary classification experiments on some widely used real-word datasets with high dimensions. We gather these data sets from the project page of LibSVM[28]. The brief descriptions of the datasets are given in Table 4.

Since we intend to conduct binary classification experiments, for MNIST, we select some pairs of digits 0-9 as a new datasets. In addition, the training dataset of rcv1.binary is much smaller than its testing dataset, so we use testing data for training and training data for testing.

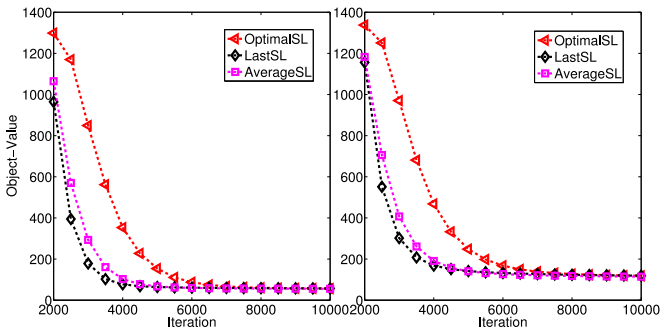


Fig. 6. The performance comparison of our three methods when d=1,000, ρ=0.1, λ=0.1 and σ_e=1 (left) and 10 (right).

5.2.2 Data Preprocessing

Some components of the data points are relatively large, like more than 100. Since there is an exponential function in our model, it would be easily to overflow if we just use the raw data directly. To address this problem, we normalize all the components of the feature vectors into the interval [0,1] before training.

5.2.3 Experimental Model and Parameter Setting

We adopt the logistic regression model in our experiments to do binary classification for its good performance in many real applications. Following the experiment design of [3], [29], we use the ℓ_1-norm regularized logistic regression model to evaluate the sparse leaning abilities of our methods. Specifically, we set the loss function as $f(\widetilde{\mathbf{w}}, z) = \log\left(1 + \exp\left(-y(\mathbf{w}^T\mathbf{x} + b)\right)\right) + \frac{\rho}{2}\|\widetilde{\mathbf{w}}\|_2^2$, where $\mathbf{w} \in \mathbb{R}^d$, b is the model bias and $\widetilde{\mathbf{w}} = [\mathbf{w}; b]$. It is straightforward to verify that $f(\widetilde{\mathbf{w}}, z)$ is a ρ-strongly convex and smooth loss function. We set the sparsity-inducing regularizer $\Psi(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$. The parameter α is set to be 0.3 for both α-SGD and our algorithms. At last, we choose different ρ and λ for different datasets to make the model reaches a good classification performance for each dataset. The concrete values of the parameters including ρ, λ can be seen in the result figures and tables.

5.2.4 Evaluation Metrics

We use objective value (Obj), test error (TE), exact density ratio, truncated density ratio (TD) and the variance of the experimental loss (Var) to evaluate the prediction models learned by different methods. The threshold of TD here is also 10^{-6}. We iterate at most 100,000 times and run each algorithm 100 times with independent random shuffles of training examples and at last report the averaged results.

To show the relevance between the selected features and our classification task, we visualize the sparse patterns of the learned prediction models of MNIST in gery-level

TABLE 4
Descriptions of the Datasets

Datasets	Dimensions	# of training	# of testing	classes
rcv1.binary	47,236	20,242	677,399	2
gisette	5,000	6,000	1,000	2
MNIST	784(28×28)	60,000	10,000	10

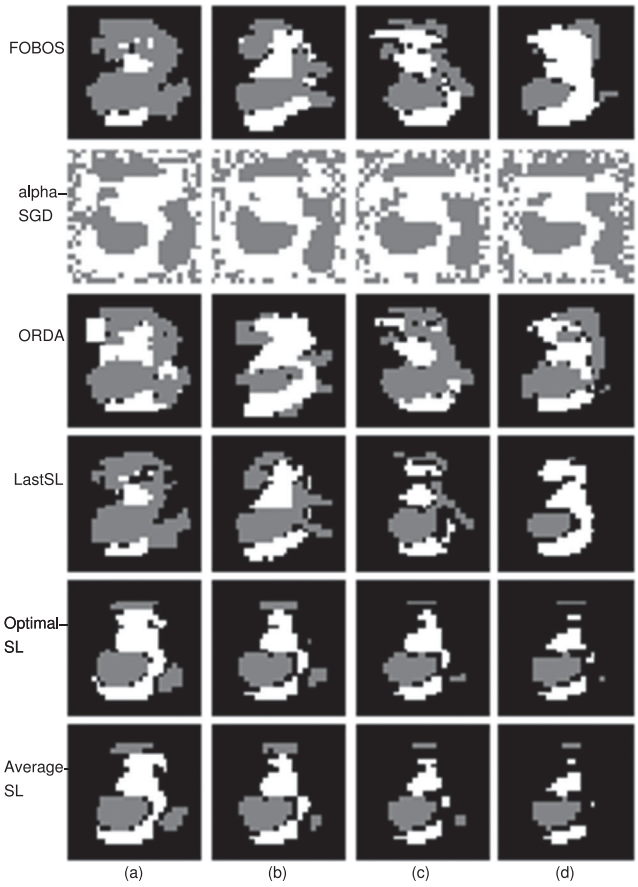


Fig. 7. The visualization for the prediction models learned by different methods when classifying the digits 2 and 3 in MNIST. Columns (a)-(d) are the results for ρ=0.01 and $\lambda = 0.02, 0.03, 0.04, 0.05$.

images. Specifically, we create a new vector $\widetilde{\mathbf{w}}'$ for a learned solution $\widetilde{\mathbf{w}}$ as follows

$$[\widetilde{\mathbf{w}}']_i = \begin{cases} 0.5 & [\widetilde{\mathbf{w}}]_i < 0 \\ 1 & [\widetilde{\mathbf{w}}]_i > 0 \\ 0 & [\widetilde{\mathbf{w}}]_i = 0. \end{cases}$$

We then reshape $\widetilde{\mathbf{w}}'$ into a matrix in the size of 28×28 and visualize it as a grey-level image. Evidently, the larger the black area in the grey-level image, the sparser the solution is. At last, we show the images for the prediction models learned by different algorithms under different parameters for classifying the digits 2 and 3 in Fig. 7.

5.2.5 Experimental Results

Due to the space limitation, we only report the results for classifying digits 2 and 3 in MNIST and rcv1.binary. All the other results can be found in the supplementary document.

On one hand, ED and TD in Tables 5 and 6 show that our methods can improve the sparsity of solutions significantly, and achieve comparable test errors. It is further confirmed by the grey-level images shown in Fig. 7, in which the solutions obtained by our methods have significantly larger black areas than those by the other algorithms. Fig. 7 also gives us a deep impression that the features selected by our methods are indeed the most relevant features for our classification task. The variances of AverageSL and OptimalSL in the Var column are much smaller than those of LastSL

TABLE 5
Results When Classify the Digits 2 and 3 of MNIST with Fixed $\rho = 0.01$

Alg	Obj	TE	ED	TD	Var	time
			$\lambda = 0.02$			
FOBOS	0.392	0.052	0.39	0.39	1.1e-04	0.34
α-SGD	0.382	0.045	1.00	0.99	2.1e-07	0.26
ORDA	0.377	0.052	0.41	0.36	5.6e-06	0.64
LastSL	**0.389**	**0.051**	**0.33**	**0.33**	**8.2e-05**	**0.27**
OptimalSL	**0.381**	**0.044**	**0.31**	**0.31**	**4.7e-07**	**0.21**
AverageSL	**0.378**	**0.045**	**0.28**	**0.28**	**1.7e-07**	**0.28**
			$\lambda = 0.03$			
FOBOS	0.462	0.057	0.38	0.38	1.7e-04	0.34
α-SGD	0.448	0.051	1.00	1.00	2.8e-07	0.26
ORDA	0.440	0.057	0.38	0.34	5.4e-06	0.64
LastSL	**0.456**	**0.056**	**0.29**	**0.29**	**1.1e-04**	**0.27**
OptimalSL	**0.444**	**0.051**	**0.24**	**0.24**	**3.6e-07**	**0.21**
AverageSL	**0.441**	**0.052**	**0.21**	**0.21**	**1.7e-07**	**0.28**
			$\lambda = 0.04$			
FOBOS	0.519	0.064	0.36	0.36	2.3e-04	0.34
α-SGD	0.502	0.055	1.00	1.00	3.7e-07	0.25
ORDA	0.491	0.061	0.36	0.32	5.6e-06	0.63
LastSL	**0.508**	**0.062**	**0.25**	**0.25**	**1.2e-04**	**0.27**
OptimalSL	**0.495**	**0.055**	**0.18**	**0.18**	**3.4e-07**	**0.21**
AverageSL	**0.492**	**0.056**	**0.16**	**0.16**	**1.4e-07**	**0.29**
			$\lambda = 0.05$			
FOBOS	0.564	0.070	0.33	0.33	3.7e-04	0.34
α-SGD	0.547	0.059	1.00	0.99	5.9e-07	0.25
ORDA	0.534	0.065	0.34	0.30	6.0e-06	0.64
LastSL	**0.550**	**0.066**	**0.21**	**0.21**	**1.5e-04**	**0.28**
OptimalSL	**0.537**	**0.059**	**0.14**	**0.14**	**3.3e-07**	**0.21**
AverageSL	**0.535**	**0.060**	**0.12**	**0.12**	**1.0e-07**	**0.28**

TABLE 6
Results of rcv1.Binary with Fixed $\rho = 0.001$

Alg	Obj	TE	ED	TD	Var	time
			$\lambda = 0.002$			
FOBOS	0.533	0.118	0.042	0.042	2.6e-07	78.8
α-SGD	0.532	0.116	1.000	0.853	2.9e-09	58.8
ORDA	0.553	0.120	0.469	0.447	7.6e-07	161.7
LastSL	**0.533**	**0.118**	**0.040**	**0.040**	**2.4e-07**	**62.7**
OptimalSL	**0.531**	**0.116**	**0.047**	**0.047**	**1.9e-09**	**46.5**
AverageSL	**0.530**	**0.116**	**0.035**	**0.035**	**7.8e-10**	**63.8**
			$\lambda = 0.003$			
FOBOS	0.576	0.147	0.034	0.034	1.4e-07	78.0
α-SGD	0.575	0.147	1.000	0.889	3.5e-09	58.9
ORDA	0.588	0.149	0.278	0.254	1.9e-06	160.1
LastSL	**0.576**	**0.147**	**0.031**	**0.031**	**1.3e-07**	**62.7**
OptimalSL	**0.573**	**0.146**	**0.027**	**0.026**	**1.7e-09**	**46.5**
AverageSL	**0.572**	**0.147**	**0.020**	**0.020**	**1.3e-09**	**63.8**
			$\lambda = 0.004$			
FOBOS	0.605	0.168	0.028	0.028	2.6e-07	77.4
α-SGD	0.604	0.168	1.000	0.901	7.7e-09	58.9
ORDA	0.609	0.173	0.151	0.128	1.2e-06	158.3
LastSL	**0.604**	**0.168**	**0.025**	**0.025**	**2.4e-07**	**62.8**
OptimalSL	**0.601**	**0.167**	**0.018**	**0.018**	**1.8e-09**	**46.5**
AverageSL	**0.600**	**0.168**	**0.013**	**0.013**	**6.6e-10**	**63.9**
			$\lambda = 0.005$			
FOBOS	0.625	0.193	0.024	0.024	5.1e-07	77.2
α-SGD	0.624	0.191	1.000	0.914	2.4e-08	58.8
ORDA	0.624	0.195	0.079	0.058	3.3e-06	157.3
LastSL	**0.624**	**0.193**	**0.021**	**0.021**	**4.7e-07**	**62.8**
OptimalSL	**0.620**	**0.191**	**0.012**	**0.012**	**2.7e-09**	**46.5**
AverageSL	**0.620**	**0.192**	**0.009**	**0.009**	**8.2e-10**	**63.9**

TABLE 7
The Test Error of α-SGD before (EB) and after (EA) Rounding When Classifying on Some Data Pairs

Digits Pair	Iterations	EB	EA	Gap
0, 5	5000	0.049	**0.073**	**0.024**
1, 5	5000	0.024	**0.039**	**0.015**
1, 8	5000	0.045	**0.057**	**0.012**
3, 5	10000	0.122	**0.135**	**0.013**
5, 8	9000	0.113	**0.127**	**0.014**

and ORDA. It may come from the lower high probability bound of AverageSL and OptimalSL.

On the other hand, we can see that among our three methods, AverageSL can find the most sparse solution with comparable objective value and test error.

At last, we investigate the performance of simple rounding by conducting an extra experiment on MNIST and give the results in Table 7. We first find a threshold for rounding that can make the rounded prediction model as sparse as the model learned by AverageSL. Then we calculate the test error of the rounded model. Form Table 7, we can observe that the simple rounding process sometimes will make the test error increase dramatically. So it is unreliable, which demonstrates our analysis in the introduction section.

6 Conclusion

In this paper, we propose a new scheme by introducing a novel sparse online-to-batch conversion procedure to the existing SO methods to learn an exactly sparse solution for SCO problems with a sparsity-inducing regularizer. Three concrete algorithms under this scheme are developed, one based on the existing SO algorithms and the other two based on SGD algorithm. We verify, both theoretically and empirically, that the proposed algorithms are significantly superior over the existing SCO methods at yielding exactly sparse solutions and they can improve the high probability bound to approximately $\mathcal{O}(\log(\log(T)/\delta)/\lambda T)$. In the future, we plan to investigate sparse online-to-batch conversion for loss functions that are only strongly convex but not necessarily smooth.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2013CB336500, National Natural Science Foundation of China under Grant 61233011, and National Youth Top-notch Talent Support Program.

References

[1] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework," *J. Soc. Indus. Appl. Math. Optim.*, vol. 22, no. 4, pp. 1469–1492, 2012.

[2] X. Chen, Q. Lin, and J. Pena, "Optimal regularized dual averaging methods for stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 404–412.

[3] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, 2010.

[4] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, 2013.

[5] W. Zhang, L. Zhang, Y. Hu, R. Jin, D. Cai, and X. He, "Sparse learning for stochastic composite optimization," in *Proc. 28th Assoc. Adv. Artif. Intell. Nat. Conf. Artif. Intell.*, 2014.

[6] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. New York, NY, USA: Springer, vol. 87, 2004.

[7] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *J. Mach. Learn. Res.*, vol. 10, pp. 777–801, 2009.

[8] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 449–456.

[9] E. Hazan and S. Kale, "Beyond the regret minimization barrier: An optimal algorithm for stochastic strongly-convex optimization," in *Proc. 24th Annu. Conf. Learn. Theory*, 2011, pp. 421–436.

[10] L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," in *Proc. Adv. Neural Inf. Processi. Syst.*, 2013, pp. 980–988.

[11] A. Agarwal, S. Negahban, and M. J. Wainwright, "Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1538–1546.

[12] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Math. Program.*, vol. 120, no. 1, pp. 221–259, 2009.

[13] Q. Lin, X. Chen, and J. Pena, "A sparsity preserving stochastic gradient method for composite optimization," *Manuscr., Carnegie Mellon Univ.*, vol. 15213, 2011.

[14] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for stochastic optimization," *J. Soc. Indus. Appl. Math. Optim.*, vol. 22, no. 2, pp. 674–701, 2012.

[15] C. Hu, W. Pan, and J. T. Kwok, "Accelerated gradient methods for stochastic optimization and online learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 781–789.

[16] Y. Nesterov, "Gradient methods for minimizing composite objective function," *Core Discuss.Pap.*, 2007

[17] G. Lan, "An optimal method for stochastic composite optimization," *Math. Program.*, vol. 133, pp. 365–397, 2012.

[18] N. Littlestone, "From on-line to batch learning," in *Proc. 2nd Annu. Workshop Comput. Learn. Theory*, 1989, pp. 269–284.

[19] O. Dekel and Y. Singer, "Data-driven online to batch conversions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 267–274.

[20] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, 2010.

[21] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2008, pp. 3869–3872.

[22] S. Becker, J. Bobin, and E. J. Candès, "Nesta: A fast and accurate first-order method for sparse recovery," *J. Soc. Indus. Appl. Math. Imaging Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[23] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, 2009.

[24] H. Oiwa, S. Matsushima, and H. Nakagawa, "Frequency-aware truncated methods for sparse online learning," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2011, pp. 533–548.

[25] S. Smale and D. Zhou, "Geometry on probability spaces," *Constructive Approx.*, vol. 30, pp. 311–323, 2009.

[26] J. A. Tropp, "Freedmans inequality for matrix martingales," *Electron. Commun. Probab.*, vol. 16, pp. 262–270, 2011.

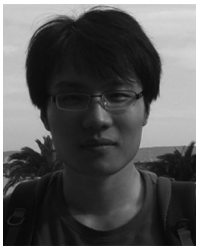
[27] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[28] C.-C. Chang and C.-J. Lin, "LibSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol*, vol. 2, no. 3, p. 27, 2011.

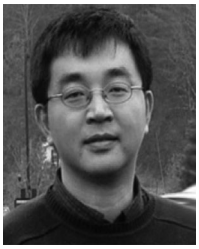
[29] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

Weizhong Zhang received the BS degree in math and applied mathematics from Zhejiang University, China, in 2012. He is currently working toward the PhD degree in computer science at Zhejiang University. His research interests include machine learning, computer vision, and data mining.

Lijun Zhang received the BS and PhD degrees in software engineering and computer science from Zhejiang University, China, in 2007 and 2012, respectively. He is currently an associate professor of the Department of Computer Science and Technology, Nanjing University, China. Prior to joining Nanjing University, he was a postdoctoral researcher at the Department of Computer Science and Engineering, Michigan State University. His research interests include machine learning, optimization, information retrieval, and data mining. He is a member of the IEEE.



Zhongming Jin received the master degree from Zhejiang University, China, in 2010. He is currently working toward the PhD degree with the State Key Laboratory of Computer-Aided Design and Computer Graphics, Zhejiang University, China. His current research interests include information retrieval, computer vision, and machine learning.

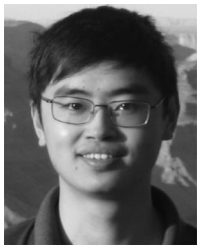


Rong Jin received the PhD degree in computer science from Carnegie Mellon University. He is a principal engineer at Alibaba group. He was a professor of Computer Science and Engineering Dept. at Michigan State University. His research is focused on statistical machine learning and its application to information retrieval. He has worked on a variety of machine learning algorithms and their application to information retrieval, including retrieval models, collaborative filtering, cross lingual information retrieval, document clustering, and video/image retrieval. He published more than 200 conference and journal articles on related topics. He received the US National Science Foundation Career Award in 2006 and supervised the Best Student Paper from COLT in 2010.

Deng Cai received the PhD degree in computer science from the University of Illinois, Urbana Champaign, in 2009. He is a professor in the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China. His research interests include machine learning, data mining, and information retrieval. He is a member of the IEEE.

Xuelong Li is a full professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P.R. China. He is a fellow of the IEEE.

Ronghua Liang received the PhD degree in computer science from Zhejiang University in 2003. He worked as a research fellow at the University of Bedfordshire, United Kingdom, from April 2004 to July 2005 and as a visiting scholar at the University of California, Davis, from March 2010 to March 2011. He is currently a professor of computer science and dean of College of Information Engineering, Zhejiang University of Technology, China. He has published more than 50 papers in leading international journals and conferences including the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Visualization and Computer Graphics*, IEEE VIS, IJCAI, AAAI. His research interests include Visual Analytics, Computer Vision, and Medical Visualization.



Xiaofei He received the BS degree in computer science from Zhejiang University, China, in 2000 and the PhD degree in computer science from the University of Chicago, in 2005. He is a professor in the State Key Lab of CAD&CG, Zhejiang University, China. Prior to joining Zhejiang University, he was a research scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision. He is a Senior member of the IEEE.