Accelerated Stochastic Gradient Method for Composite Regularization

Leon Wenliang Zhong **James T. Kwok**
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Abstract

Regularized risk minimization often involves nonsmooth optimization. This can be particularly challenging when the regularizer is a sum of simpler regularizers, as in the overlapping group lasso. Very recently, this is alleviated by using the *proximal average*, in which an implicitly nonsmooth function is employed to approximate the composite regularizer. In this paper, we propose a novel extension with accelerated gradient method for stochastic optimization. On both general convex and strongly convex problems, the resultant approximation errors reduce at a faster rate than methods based on stochastic smoothing and ADMM. This is also verified experimentally on a number of synthetic and real-world data sets.

1 Introduction

Regularized risk minimization is a fundamental tool in machine learning. It admits a tradeoff between the empirical loss and regularization, as

$$\min_{\mathbf{x}} \frac{1}{n}\sum_{t=1}^{n} \ell(\mathbf{x}; \mathbf{s}_t, l_t) + r(\mathbf{x}). \tag{1}$$

Here, $\mathbf{x} \in \mathbb{R}^d$ is the model parameter, n is the number of samples, $\ell(\mathbf{x}; \mathbf{s}_t, l_t)$ is the empirical loss on sample t with input $\mathbf{s}_t$ and output l_t, and $r(\mathbf{x})$ is a regularizer on $\mathbf{x}$. In this paper, we will only focus on convex losses and convex regularizers. A number of optimization tools have been proposed for solving (1). Among these, an important family is the gradient descent. It uses only first-order information, and is easy to implement and highly scalable. When both the loss and regularizer are smooth, first-order methods can be accelerated by Nesterov's optimal approach [12]. In particular, it enjoys a convergence rate of $\mathcal{O}\left(\frac{1}{T^2}\right)$, where T is the number of iterations. Equivalently, an ϵ-accurate solution can be obtained in $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$ iterations.

Many modern learning models involve nonsmooth components. For example, in the SVM, the hinge loss is nonsmooth and allows for sparse support vectors; in lasso [19], its nonsmooth ℓ_1-regularizer performs automatic feature selection during learning. More examples can be found in [1]. However, gradient descent becomes less appealing when nonsmooth components are present in the optimization objective. Indeed, a direct extension using the subgradient is often criticized for its slow convergence rate of $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ [11].

In recent years, more sophisticated optimization algorithms have been developed to handle nonsmooth objectives. When only the regularizer is nonsmooth, the (accelerated) proximal methods [5, 14] employ a linear approximation at the current solution estimate, while leaving the nonsmooth term (typically the regularizer) intact. This enjoys the optimal convergence rate as for smooth problems. An essential building block in each of its iterations is the proximal step

$$\mathrm{M}_r^{\eta}(\mathbf{x}) = \min_{\mathbf{y}} \frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\eta} + r(\mathbf{y}), \tag{2}$$

where $\eta > 0$. Though this step can often be efficiently solved for "simple" regularizers [7], it becomes more challenging in problems such as the generalized lasso [20] and overlapping group lasso [24], in which $r(\mathbf{x})$ is a composite regularizer of the form $\sum_{k=1}^{K} w_k r_k(\mathbf{x})$ for some $w_k \geq 0$ and convex $r_k(\mathbf{x})$'s. Pioneering works [3, 10] often convert this proximal step to its dual, which is then solved with nonlinear optimization (such as the network flow algorithm or Newton's method). However, this approach is difficult to generalize as the dual is highly problem-dependent. Moreover, despite a faster theoretical convergence rate, it may be even slower than stochastic gradient descent in practice [21].

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

To handle a combination of regularizers, a useful strategy is "divide-and-conquer". In particular, methods based on the alternating direction method of multipliers (ADMM) [6] duplicate the model parameter in each regularizer, and enforce the duplicates to be identical with equality constraints. It can be shown that updating these duplicated variables is equivalent to solving the proximal step for each regularizer separately [6]. Nesterov's smoothing technique [13], which first approximates the nonsmooth term with a smooth function and then solve the resultant smoothed problem with accelerated gradient method, is also applicable in this scenario. As pointed out in [23], the most popular approximation, which adds a quadratic function to each regularizer, is indeed computing a proximal step to replace the original regularizer.

Very recently, Yu [23] introduced the proximal average technique for proximal gradient methods (PA-APG). Instead of directly solving the proximal step associated with a composite regularizer, it averages the solutions from the proximal problems for each regularizer. It is shown that this approximation is strictly better than that of Nesterov's smoothing, while enjoying the same per-iteration time complexity and convergence rate.

In the context of regularized risk minimization, the deterministic setting corresponds to batch learning, and each iteration needs to visit all the training samples. With the proliferation of data-intensive applications, this can quickly become computationally infeasible. To alleviate this problem, stochastic techniques have recently drawn a lot of interest. Most are based on (variants of) the stochastic gradient descent (SGD). Recently, a stochastic variant of the smoothing technique is developed in [15], while stochastic versions of the ADMM are proposed in [16, 18]. However, for the proximal average technique in [23], how to extend it for the stochastic setting, together with its theoretical analysis and empirical performance, still remain open. Besides, Yu [23] does not take strong convexity into consideration, though it is well-known that it can often speed up first-order methods.

In this paper, we develop a stochastic accelerated gradient algorithm based on the proximal average. It will be shown that the proposed algorithm has a convergence rate of $\mathcal{O}\left(\frac{1}{T^2}+\frac{1}{T^{\frac{3}{2}}}+\frac{1}{T}+\frac{1}{\sqrt{T}}\right)$ on general convex problems, and a $\mathcal{O}\left(\frac{1}{T^2}+\frac{\log T}{T^2}+\frac{1}{T}\right)$ rate on strongly convex problems. Here, the $\mathcal{O}\left(\frac{1}{T^{\frac{3}{2}}}\right)$ and $\mathcal{O}\left(\frac{\log T}{T^2}\right)$ terms are due to the use of proximal average, and are faster than the $\mathcal{O}\left(\frac{1}{T}\right)$ rate for ADMM-based and stochastic smoothing methods [15, 16, 18].

The rest of this paper is organized as follows. Section 2 introduces the problem formulation and gives brief reviews on accelerated gradient methods, Nesterov's smoothing technique and the proximal average. Section 3 then describes the proposed algorithm. Experimental results are presented in Section 4, and the last section gives some concluding remarks.

Notation. In the sequel, the transpose of vector/matrix is denoted by the superscript ^{T}, and $\|\mathbf{x}\|$ denotes the Euclidean norm of a vector $\mathbf{x}$. For a differentiable function f, we use ∇f for its gradient.

2 Background and Related Work

2.1 Problem Formulation

We consider the following stochastic optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \phi(\mathbf{x}) \equiv \mathbb{E}_\xi[f(\mathbf{x},\xi)] + \mathbb{E}_\xi[g(\mathbf{x},\xi)] + r(\mathbf{x}), \quad (3)$$

where ξ is a random variable, and the three components on the RHS satisfy the following assumptions:

A1 $f(\mathbf{x}) \equiv \mathbb{E}_\xi[f(\mathbf{x},\xi)]$ is μ-strongly convex (where $\mu \geq 0$) with L_f-Lipschitz continuous gradient. In other words, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\begin{aligned} f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{x}-\mathbf{y}\|^2 &\leq f(\mathbf{y}), \\ f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{L_f}{2}\|\mathbf{x}-\mathbf{y}\|^2 &\geq f(\mathbf{y}). \end{aligned}$$

A2 $g(\mathbf{x}) \equiv \mathbb{E}_\xi[g(\mathbf{x},\xi)]$ is convex but nonsmooth. Moreover, for a given ξ, $g(\mathbf{x},\xi)$ can be written as

$$g(\mathbf{x},\xi) = \max_{\mathbf{y}\in\Omega} (\mathbf{A}_\xi \mathbf{x})^T\mathbf{y} - Q(\mathbf{y}), \quad (4)$$

where Q is convex and continuous, $\mathbf{A}_\xi$ is a matrix associated with ξ, and Ω is a convex set. This particular structure will be useful in applying Nesterov's smoothing technique [13] (Section 2.3).

A3 r is a convex combination of convex functions $r_1, r_2, \ldots, r_K$, i.e.,

$$r(\mathbf{x}) = \sum_{k=1}^K w_k r_k(\mathbf{x}), \quad (5)$$

where $w_k \geq 0$ and $\sum_{k=1}^K w_k = 1$. Each r_k is possibly nonsmooth but assumed to be L_{r_k}-Lipschitz continuous, i.e., $|r_k(\mathbf{x}) - r_k(\mathbf{y})| \leq L_{r_k}\|\mathbf{x}-\mathbf{y}\|$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

In the context of regularized risk minimization, $\mathbf{x}$ is the model parameter to be learned, f and g are the

empirical losses, and r_k's the regularizers. While most machine learning models have only one loss term, here we have two as this gives added flexibility. Besides, as will be seen, they have different contributions to the convergence rate. Depending on the application, one can set $f = 0$ or $g = 0$.

The following shows the forms of $r(\mathbf{x})$ in (5) for some popular machine learning models.

- Overlapping group lasso [10, 24]: Here, the model parameter $\mathbf{x}$ is divided into K possibly overlapping groups. Let group k be $\mathbf{g}_k \subseteq \{1, 2, \ldots, d\}$, and the corresponding subvector of $\mathbf{x}$ be $\mathbf{x}_{\mathbf{g}_k}$. In (5), $r_k(\mathbf{x}) = \|\mathbf{x}_{\mathbf{g}_k}\|_p$ (where $p = 2$ or ∞), and w_k is the (normalized) weight for group k. This regularizer can be used to select groups of features.

- Graph-guided fused lasso [15]: Here, features are represented as vertices in a graph, and related features are connected by edges. K in (5) is then the number of edges, w_k is the (normalized) weight for edge k, and $r_k(\mathbf{x}) = |x_{k_1} - x_{k_2}|$ where k_1, k_2 are features connected by edge k. This regularizer encourages coefficients of highly related features to be similar to each other.

- Sparse and low-rank matrix estimation [17]: In this case, $\mathbf{x}$ is a matrix, and a combination of the ℓ_1-regularizer ($r_1(\mathbf{x}) = \sum_{ij} |x_{ij}|$) and nuclear norm regularizer ($r_2(\mathbf{x}) = \|\mathbf{x}\|_*$, the sum of $\mathbf{x}$'s singular values) encourages the solution to be simultaneously sparse and low-rank. Thus, K in (5) is 2, and w_1, w_2 are the tradeoff parameters.

2.2 Accelerated Gradient Methods

Though highly scalable, gradient descent is often criticized for its slow convergence rate. Nesterov pioneered the accelerated gradient descent (AGD) method for smooth optimization, which achieves the optimal convergence rate for a black-box model [12]. Later, it is extended to accelerated proximal gradient (APG) [5, 14] for composite optimization problems, in which the objective contains a smooth term $f(\mathbf{x})$ and a nonsmooth term $r(\mathbf{x})$. In each iteration t, a quadratic function is used to upper bound the smooth $f(\mathbf{x})$, while leaving the nonsmooth $r(\mathbf{x})$ intact, leading to the optimization problem

$$\min_{\mathbf{x}} f(\mathbf{y}_t) + \nabla f(\mathbf{y}_t)^T(\mathbf{x} - \mathbf{y}_t) + \frac{\|\mathbf{x} - \mathbf{y}_t\|^2}{2\eta_t} + r(\mathbf{x}), \quad (6)$$

where η_t is the stepsize, and $\mathbf{y}_t$ is a linear combination of the last two estimates $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. It is easy to see that (6) can be converted to the proximal step $\mathrm{M}_r^\eta(\mathbf{x})$ in (2). For APG to be effective, the nonsmooth $r(\mathbf{x})$ has to be "simple", i.e., the corresponding $\mathrm{M}_r^\eta(\mathbf{x})$ needs to allow efficient computation.

Both AGD and APG enjoy the optimal convergence rate of $\mathcal{O}\left(\frac{1}{T^2}\right)$. Recently, several stochastic extensions are introduced [8, 22], which have a convergence rate of $\mathcal{O}\left(\frac{1}{T^2} + \frac{1}{\sqrt{T}}\right)$. Here, the extra $\mathcal{O}(\frac{1}{\sqrt{T}})$ term is related to the variance of the stochastic gradients. While these stochastic accelerated variants may be as slow as the simple stochastic gradient methods when the variance is large, this can be alleviated by reducing the variance with the use of mini-batch [8].

2.3 Smoothing Nonsmooth Functions

As discussed in Section 1, nonsmooth functions are more difficult to optimize than smooth functions. Nesterov [13] showed that a smooth approximation with Lipschitz-continuous gradient can be obtained when the nonsmooth component, denoted $h(\mathbf{x})$, is of the form

$$h(\mathbf{x}) = \max_{\mathbf{y} \in \Omega} (\mathbf{A}\mathbf{x})^T\mathbf{y} - Q(\mathbf{y}), \quad (7)$$

where Q is convex, and Ω is a convex set. Specifically, let ω be a ζ-strongly convex function, and define

$$\hat{h}(\mathbf{x}) = \max_{\mathbf{y} \in \Omega} (\mathbf{A}\mathbf{x})^T\mathbf{y} - Q(\mathbf{y}) - \gamma\omega(\mathbf{y}), \quad (8)$$

where $\gamma > 0$ is constant. It can be shown that $\hat{h}(\mathbf{x})$ is convex and its gradient $\nabla\hat{h}(\mathbf{x}) = \mathbf{A}^T\mathbf{y}(\mathbf{x})$ is $\frac{\|\mathbf{A}\|^2}{\gamma\zeta}$-Lipschitz continuous, where $\mathbf{y}(\mathbf{x})$ is the optimal $\mathbf{y}$ (for the given $\mathbf{x}$) in (8), and $\|\mathbf{A}\| = \max_{\mathbf{x},\mathbf{y}} (\mathbf{A}\mathbf{x})^T\mathbf{y}$: $\|\mathbf{x}\| = \|\mathbf{y}\| = 1$. Interestingly, when $h(\mathbf{x})$ is convex and Lipschitz-continuous, Yu [23] showed that it can always be written in the form of (7), with $\mathbf{A} = \mathbf{I}$ and Ω being the domain of the Fenchel conjugate of $h(\mathbf{x})$. On using $\omega(\mathbf{y}) = \frac{1}{2}\|\mathbf{y}\|^2$, it can be further shown that $\hat{h}(\mathbf{x})$ is the same as $\mathrm{M}_h^\gamma(\mathbf{x})$ in (2).

The following Lemma shows that the smooth surrogate $\hat{h}(\mathbf{x})$ is close to the original $h(\mathbf{x})$ with a sufficiently small γ. With a carefully choosing γ, AGD on $\hat{h}(\mathbf{x})$ has a convergence rate of $\mathcal{O}\left(\frac{1}{T}\right)$ [13].

Lemma 1 *[13]* $0 \leq h(\mathbf{x}) - \hat{h}(\mathbf{x}) \leq \gamma D_\Omega$, *where* $D_\Omega = \max_{\mathbf{y} \in \Omega} \omega(\mathbf{y})$.

As an application, note that the composite regularizer $r(\mathbf{x})$ in (5) is typically nonsmooth and thus smoothing can be used. Specifically, as all the r_k's in (5) are convex and Lipschitz-continuous, each of them can be smoothed separately, leading to the smooth approximation

$$r(\mathbf{x}) \simeq \sum_{k=1}^{K} w_k \mathrm{M}_{r_k}^\gamma(\mathbf{x}). \quad (9)$$

Figure 1: Accelerated Stochastic Gradient Descent with Proximal Average (PA-ASGD).

1: Input: Sequences $\{\mathcal{L}_t\}$, $\{\alpha_t\}$, $\{\eta_t\}$ and $\{\gamma_t\}$.
2: Initialize: $\bar{\mathbf{y}}_{-1} = \mathbf{z}_{-1} = \mathbf{0}$, $\alpha_0 = \lambda_0 = 1$.
3: **for** $t = 0, \ldots$ **do**
4: $\quad \mathbf{x}_t = \left[(1-\alpha_t)(\mu + \mathcal{L}_t\alpha_t)\bar{\mathbf{y}}_{t-1} + \mathcal{L}_t\alpha_t^2\mathbf{z}_{t-1}\right] / \left[\mu(1-\alpha_t) + \mathcal{L}_t\alpha_t\right]$;
5: $\quad \mathbf{y}_t = \mathbf{x}_t - \eta_t\left[\nabla f(\mathbf{x}_t, \xi_t) + \nabla\hat{g}(\mathbf{x}_t, \xi_t)\right]$;
6: $\quad \bar{\mathbf{y}}_t = \sum_{k=1}^K w_k \mathrm{P}_{r_k}^{\eta_t}(\mathbf{y}_t)$;
7: $\quad \mathbf{z}_t = \mathbf{z}_{t-1} - \left[\mathcal{L}_t(\mathbf{x}_t - \bar{\mathbf{y}}_t) + \mu(\mathbf{z}_{t-1} - \mathbf{x}_t)\right] / \left[\mathcal{L}_t\alpha_t + \mu\right]$;
8: **end for**
9: Output $\bar{\mathbf{y}}_t$.

Recently, this smoothing technique is extended to the stochastic setting [15]. When the nonsmooth $g(\mathbf{x}, \xi)$ is of the form in (4) (such as the hinge loss [15]), it can be approximated by the smooth function

$$\hat{g}(\mathbf{x}, \xi) = \max_{\mathbf{y}\in\Omega} (\mathbf{A}_\xi\mathbf{x})^T\mathbf{y} - Q(\mathbf{y}) - \gamma\omega(\mathbf{y}), \qquad (10)$$

with Lipschitz-continuous gradient

$$\nabla\hat{g}(\mathbf{x}, \xi) = \mathbf{A}_\xi^T\mathbf{y}_\xi(\mathbf{x}), \qquad (11)$$

where $\mathbf{y}_\xi(\mathbf{x})$ is the optimal $\mathbf{y}$ in (10). Analogous to Lemma 1, $\hat{g}(\mathbf{x}) = \mathbb{E}[\hat{g}(\mathbf{x}, \xi)]$ is close to $g(\mathbf{x}) = \mathbb{E}[g(\mathbf{x}, \xi)]$ with a sufficiently small γ.

Lemma 2 *[15]* $0 \le g(\mathbf{x}) - \hat{g}(\mathbf{x}) \le \gamma D_\Omega$.

2.4 Proximal Average

As mentioned in Section 1, the proximal step (2) is a fundamental building block in proximal gradient methods. Its use with simple regularizers (such as $r(\mathbf{x}) = \|\mathbf{x}\|_1$ or $\|\mathbf{x}\|_\infty$) has been extensively studied in the literature [7]. However, when $r(\mathbf{x})$ is a combination of regularizers as in (5), efficient solutions for $\mathrm{M}_r^\eta(\mathbf{x})$ are often hard to obtain. Recently, this problem is tackled with the introduction of the proximal average [23]. Interestingly, it is closely connected to the smoothing technique but strictly better.

Definition 1 (Proximal Average) *[4, 23] Let $P_r^\eta(\mathbf{x}) = \arg M_r^\eta(\mathbf{x})$, where $M_r^\eta(\mathbf{x})$ is as defined in (2). Given functions $\{r_k\}_{k=1}^K$ and weights $\{w_k\}_{k=1}^K$, where $w_k \ge 0$ and $\sum_{k=1}^K w_k = 1$, for a fixed $\eta > 0$, there exists an unique convex function $\hat{r}$, called the* proximal average*, such that*

$$M_{\hat{r}}^\eta(\mathbf{x}) = \sum_{k=1}^K w_k M_{r_k}^\eta(\mathbf{x}),\ P_{\hat{r}}^\eta(\mathbf{x}) = \sum_{k=1}^K w_k P_{r_k}^\eta(\mathbf{x}). \qquad (12)$$

The following Lemma bounds its approximation error with $r(\mathbf{x})$ [23].

Lemma 3 $0 \le r(\mathbf{x}) - \hat{r}(\mathbf{x}) \le \frac{\eta\bar{L}^2}{2}$, *where* $\bar{L}^2 = \sum_{k=1}^K w_k L_{r_k}^2$.

Recall from (9) that $r(\mathbf{x})$ can be replaced by $\sum_{k=1}^K w_k \mathrm{M}_{r_k}^\eta(\mathbf{x})$, which is the same as $\mathrm{M}_{\hat{r}}^\eta(\mathbf{x})$ on using (12). It can be shown that as an approximation of $r(\mathbf{x})$, the proximal average $\hat{r}(\mathbf{x})$ is at least as good as the smooth surrogate $\mathrm{M}_{\hat{r}}^\eta(\mathbf{x})$, i.e., $\mathrm{M}_{\hat{r}}^\eta(\mathbf{x}) \le \hat{r}(\mathbf{x}) \le r(\mathbf{x})$ [23].

In [23], $\mathrm{M}_{\hat{r}}^\eta(\mathbf{x})$ is used to replace $\mathrm{M}_r^\eta(\mathbf{x})$. If all the r_k's are simple regularizers, $\mathrm{M}_{r_k}^\eta(\mathbf{x})$'s and $\mathrm{P}_{r_k}^\eta(\mathbf{x})$'s can be easily computed, and subsequently so are $\mathrm{M}_{\hat{r}}^\eta(\mathbf{x})$ and $\mathrm{P}_{\hat{r}}^\eta(\mathbf{x})$ by (12). It can be easily seen that using this substituted proximal step is the same as running the proximal gradient (with a stepsize of η) on the nonsmooth surrogate $f(\mathbf{x}) + \hat{r}(\mathbf{x})$. With a suitable η, a convergence rate of $\mathcal{O}(\frac{1}{T})$ can be obtained. Though this is only the same as applying smoothing on $r(\mathbf{x})$ (Section 2.3), using proximal average is strictly better than smoothing when the constant factor inside the $\mathcal{O}(\cdot)$ is taken into consideration [23].

3 Accelerated Stochastic Gradient with Proximal Average

In this section, we combine the techniques of accelerated gradient, smoothing and proximal average to solve the stochastic optimization problem in (3).

3.1 Proposed Algorithm

The whole procedure is shown in Figure 1, and will be called "accelerated stochastic gradient descent with proximal average" (PA-ASGD) in the sequel. As in Section 2.3, the nonsmooth component $g(\mathbf{x}, \xi)$ in the objective is replaced by the smooth approximation $\hat{g}(\mathbf{x}, \xi)$ in (10). Similar to other accelerated stochastic gradient methods [8, 15], we maintain three sequences of variables $\{\mathbf{x}_t\}, \{\bar{\mathbf{y}}_t\}$ and $\{\mathbf{z}_t\}$. Sequences $\{\mathcal{L}_t\}$, $\{\alpha_t\}$ and $\{\eta_t\}$ are used to control the convergence rate; while sequence $\{\gamma_t\}$ controls the approximation quality of $\hat{g}(\mathbf{x}, \xi)$ according to Lemma 2. Their settings on

general and strongly convex problems will be specified in Section 3.3.

In Figure 1, Step 5 performs a gradient descent step on the smooth surrogate $f(\mathbf{x}_t, \xi_t) + \hat{g}(\mathbf{x}_t, \xi_t)$, where η_t is the stepsize at iteration t. Extension to the use of a mini-batch is straightforward. In a standard proximal gradient algorithm, the next step will involve computing the proximal step associated with the composite regularizer $r(\mathbf{x})$. Here, Step 6 instead computes the solution of the proximal step $\mathrm{P}^{\eta_t}_{\hat{r}_t}(\mathbf{y}_t)$, where $\hat{r}_t$ is the proximal average of the r_k's with $\eta = \eta_t$. As discussed in Section 2.4, this is much easier when the individual r_k's are simple but not their weighted combination r. Finally, it can be shown that Steps 4 and 7 together make $\mathbf{x}_t$ a combination of $\bar{\mathbf{y}}_{t-1}$ and $\bar{\mathbf{y}}_{t-2}$. This can be seen clearly when $\mu = 0$, when we then have

$$\mathbf{x}_t = ((1-\alpha_t)\bar{\mathbf{y}}_{t-1} + \alpha_t \bar{\mathbf{y}}_{t-2}) + \frac{\alpha_t}{\alpha_{t-1}}(\bar{\mathbf{y}}_{t-1} - \bar{\mathbf{y}}_{t-2}).$$

Thus, similar to AGD and APG [12, 5], the proposed PA-ASGD algorithm performs proximal gradient descent based on a combination of the estimates obtained in the previous two iterations.

3.2 Discussion

Note that, in contrary, a constant stepsize η is used in PA-APG. However, in stochastic optimization solvers, η often decreases with t (typically as $\mathcal{O}(t^{\frac{3}{2}})$ for general convex problems, and $\mathcal{O}(t^2)$ for strongly convex problems [8, 9]). The use of a decreasing $\{\eta_t\}$ sequence is also beneficial with our use of the proximal average, as $\{\hat{r}_t(\mathbf{x})\}$ becomes closer and closer to the original composite regularizer $r(\mathbf{x})$ according to Lemma 3.

The most expensive steps in Figure 1 are

- Step 5, which computes $\nabla\hat{g}(\mathbf{x}_t, \xi_t)$ using (11). In turn, this involves solving (10).
- Step 6, which computes the proximal averages of all the $\mathrm{P}^{\eta_t}_{r_k}(\mathbf{y}_t)$'s.

In many applications, solving (10) and computing each $\mathrm{P}^{\eta_t}_{r_k}(\mathbf{y}_t)$ take time (almost) linear in d (the dimension of $\mathbf{x}$) [7, 13, 15, 23]. For example, this is the case when $g(\mathbf{x}_t, \xi_t)$ is the hinge loss and the r_k's are simple. The time complexity of each PA-ASGD iteration is then $\mathcal{O}(d(n_b + K))$, where n_b is the mini-batch size. It is the same as methods based on smoothing and ADMM-based methods [18, 16], as they also have to compute the proximal step $\mathrm{M}^{\eta_t}_{r_k}(\cdot)$.

3.3 Convergence Analysis

In this section, we study the convergence rates of PA-ASGD on both general convex and strongly convex problems. We assume that the stochastic gradient $\nabla f(\mathbf{x}_t, \xi_t)$ (resp. $\nabla\hat{g}(\mathbf{x}_t, \xi_t)$) is an unbiased estimator of $\nabla f(\mathbf{x}_t)$ (resp. $\nabla\hat{g}(\mathbf{x}_t)$), i.e., $\mathbb{E}_{\xi_t}[\nabla f(\mathbf{x}_t, \xi_t)] = \nabla f(\mathbf{x})$ and $\mathbb{E}_{\xi}[\nabla\hat{g}(\mathbf{x}_t, \xi_t)] = \nabla\hat{g}(\mathbf{x}_t)$. Moreover, we assume that $\mathbb{E}[\|\mathbf{x}^* - \mathbf{z}_t\|^2] \le D^2$ for some constant D, where $\mathbf{x}^*$ is the optimal solution of (3).

3.3.1 General Convex Problems

Theorem 1 *For $t \ge 0$, on setting*

$$\begin{cases} \mathcal{L}_t = b(t+1)^{\frac{3}{2}} + L_f + \frac{\mathbb{E}\|\mathbf{A}_\xi\|^2}{\gamma_t \zeta}, \\ \alpha_t = \frac{2}{t+2}, \\ \gamma_t = \alpha_t, \\ \eta_t = \frac{1}{\mathcal{L}_t}, \end{cases} \tag{13}$$

where $b > 0$ is a constant, and ζ, D_Ω are as defined in Section 2.3, the expected error of the PA-ASGD solution can be bounded as

$$\mathbb{E}[\phi(\bar{\mathbf{y}}_T)] - \phi(\mathbf{x}^*) \le \frac{C_1}{T^2} + \frac{C_2}{T^{\frac{3}{2}}} + \frac{C_3}{T} + \frac{C_4}{\sqrt{T}}, \tag{14}$$

where

$$\begin{aligned} C_1 &= 3L_f D^2, \\ C_2 &= 4\bar{L}^2 b^{-1}, \\ C_3 &= 2\mathbb{E}[\|\mathbf{A}_\xi\|^2]\zeta^{-1}D^2 + 4D_\Omega, \\ C_4 &= 3D^2 b + 2\sigma^2 b^{-1}, \end{aligned}$$

and σ is an upper bound of the stochastic noise.

The RHS of (14) has four components. The first one comes from the smooth component f in the objective, and the others are errors introduced by the proximal average on $r(\mathbf{x})$, smoothing[1] of $g(\mathbf{x}, \xi)$ and the variance of the stochastic gradient, respectively.

For the special case where $g \equiv 0$, the bound in (14) reduces to $\mathcal{O}\left(\frac{1}{T^2} + \frac{1}{T^{\frac{3}{2}}} + \frac{1}{\sqrt{T}}\right)$. For comparison, if we instead apply smoothing on $r(\mathbf{x})$ (i.e., approximate it as $\sum_{k=1}^K w_k \mathrm{M}^{\gamma}_{r_k}(\mathbf{x})$) in (9), and then run ANSGD [15] or PA-ASGD, the convergence rate degenerates to $\mathcal{O}\left(\frac{1}{T^2} + \frac{1}{T} + \frac{1}{\sqrt{T}}\right)$. Hence, PA-ASGD can decrease the error faster.

When both $g \equiv 0$ and $\sigma = 0$ (i.e., regularized batch learning problems with a smooth loss), setting $b = 0$ in (13) eliminates the slowest $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ term. However, it also make C_2 in (14) go to infinity. Interestingly, it can be shown that if we change $\mathcal{L}_t$ in (13) to $L_f + \frac{\tilde{b}}{\gamma_t}$, where $\tilde{b} > 0$, PA-ASGD converges at the rate of $\mathcal{O}\left(\frac{1}{T^2} + \frac{1}{T}\right)$.

[1] As expected, the $\mathcal{O}(\frac{1}{T})$ rate here is the same as in [15], though with a slightly different constant.

Note that PA-APG uses a fixed stepsize η, which has to be very small when a good approximation is desired [23]. On the other hand, our η_t decreases with t and is not directly tied to the desired approximation quality. Hence, PA-ASGD may employ a more aggressive stepsize at the beginning, and thus be faster than PA-APG. This will be empirically verified in Section 4.4.

3.3.2 Strongly Convex Problems

When the problem is strongly convex ($\mu > 0$), the following Theorem shows that the convergence rate can be further improved.

Theorem 2 *Assume the same conditions as in Theorem 1. Set*

$$\begin{cases} \mathcal{L}_t = L_f + \frac{\mathbb{E}\|\mathbf{A}_\xi\|^2}{\gamma_t \zeta} + \frac{\mu}{2\alpha_t^2} - \frac{\mu}{\alpha_t}, \\ \alpha_0 = 1 \text{ and } \alpha_t = \frac{2}{t+1}, t \geq 1, \\ \gamma_t = \alpha_t, \\ \eta_t = \left(\mathcal{L}_t + \frac{\mu}{\alpha_t}\right)^{-1}. \end{cases} \tag{15}$$

The expected error of PA-ASGD can be bounded as

$$\mathbb{E}[\phi(\bar{\mathbf{y}}_T)] - \phi(\mathbf{x}^*) \leq \frac{\tilde{C}_1}{T^2} + \frac{\tilde{C}_2 \log T}{T^2} + \frac{\tilde{C}_3}{T}, \tag{16}$$

where

$$\begin{aligned} \tilde{C}_1 &= (12L_f + 4(\tilde{C}_4 + 1)\mathbb{E}\|\mathbf{A}_\xi\|^2 \zeta^{-1})D^2 + 8\bar{L}^2, \\ \tilde{C}_2 &= 8\bar{L}^2\mu^{-1}, \\ \tilde{C}_3 &= 3D_\Omega + 4\sigma^2\mu^{-1}, \\ \tilde{C}_4 &= \max\left\{2\left(\frac{L_f}{\mu}\right)^{\frac{1}{3}}, \frac{4\mathbb{E}\|\mathbf{A}_\xi\|^2}{\zeta\mu}\right\}. \end{aligned}$$

For the three terms in (16), the first one comes from a combination of the smooth component f in the objective, smoothing and proximal average; the second one from the proximal average on $r(\mathbf{x})$, and the last one is due to errors introduced by smoothing of $g(\mathbf{x}, \xi)$ and variance of the stochastic gradient. As can be seen, the term due to proximal average converges at a faster rate ($\mathcal{O}\left(\frac{\log T}{T^2}\right)$) than that of smoothing ($\mathcal{O}\left(\frac{1}{T}\right)$ rate).

In the batch setting, one can remove the $4\sigma^2\mu^{-1}$ term from $\tilde{C}_3$. The other parameters ($\mathcal{L}_t, \eta_t, \alpha_t$ and γ_t) are independent of the noise in the stochastic gradient.

4 Experiments

In this section, we perform experiments on the (general convex) overlapping group lasso and (strongly convex) graph-guided logistic regression models, under both the stochastic and batch settings.

4.1 Setup

The following methods are compared:

1. the proposed PA-ASGD, which uses the settings in (13) for general convex problems, and (15) for strongly convex problems;
2. OPG-ADMM [18]: stochastic ADMM based on stochastic gradient descent;
3. RDA-ADMM [18]: stochastic ADMM based on regularized dual average [22];
4. ANSGD [15]: accelerated stochastic gradient descent with Nesterov's smoothing technique;
5. PA-PG [23]: deterministic gradient descent with proximal average; and
6. PA-APG [23]: deterministic accelerated gradient descent with proximal average.

Note that PA-PG and PA-APG can only be used in the batch setting.

For the models considered here (overlapping group lasso and graph-guided logistic regression), both the gradient and proximal step $\mathrm{M}^{\eta}_{r_k}(\cdot)$ can be computed in time linear in the dimension d. Hence, all the algorithms above have the same per-iteration complexity. Consequently, we compare their performance only in terms of the number of iterations. Moreover, we do not compare with other stochastic algorithms such as SGD, as they have been shown to be inferior [15, 18].

Each algorithm has some free data-dependent parameter(s) (such as b in (13)). To tune these, we use a small training subset, and choose the parameter setting with the smallest training objective value after running the stochastic algorithm for 200 iterations.

All methods are implemented in MATLAB. Experiments are performed on a PC with Intel i7-2600K CPU and 32GB memory. To reduce statistical variability, results are averaged over 10 repetitions.

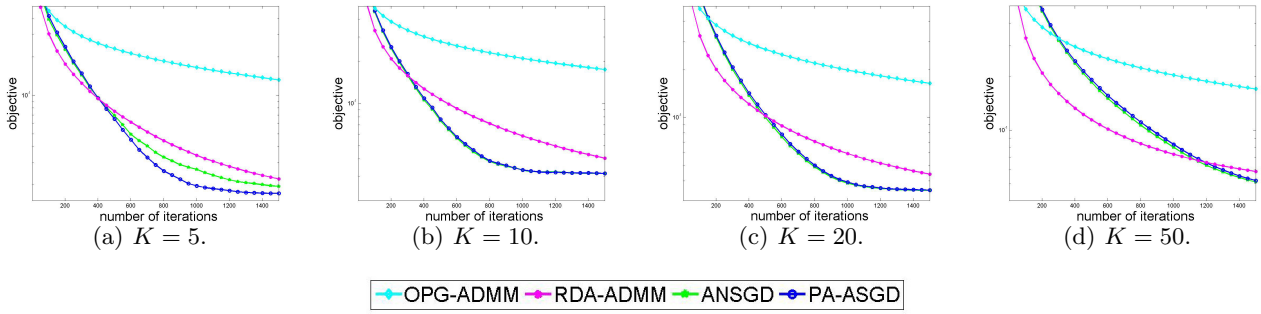
4.2 Overlapping Group Lasso

We first perform experiments on the overlapping group lasso model [24] with the hinge loss:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}[1 - l_i\mathbf{x}^T\mathbf{s}_i]_+ + \lambda\sum_{k=1}^{K}\|\mathbf{x}_{\mathbf{g}_k}\|,$$

where $[a]_+ = a$ if $a \geq 0$; and 0 otherwise. Similar to [23], we set the ground truth $\mathbf{x}^*$ as $x_j^* = (-1)^j \exp(-\frac{j-1}{100})$, and the groups are defined as

$$\underbrace{\{1, \ldots, 100\}, \{91, \ldots, 190\}, \ldots, \{d-99, \ldots, d\}}_{K \text{ groups}},$$



(a) $K = 5$. (b) $K = 10$. (c) $K = 20$. (d) $K = 50$.

Figure 2: Objective value versus number of iterations on stochastic overlapping group lasso.

where $d = 90K + 10$. The input $\mathbf{s}_i \in \mathbb{R}^d$ of each sample is generated i.i.d. from the normal distribution $\mathcal{N}(0, 1)$. Its output l_i is set to 1 if $\mathbf{x}^{*T}\mathbf{s}_i + \vartheta_i \geq 0$ (where $\vartheta_i \sim \mathcal{N}(0, 1)$); and -1 otherwise. We set $\lambda = K/(5n)$, and vary K in $\{5, 10, 20, 50\}$. Moreover, $n = d$ and the mini-batch size is $n_b = n/10$.

For PA-ASGD and ANSGD, the hinge loss is smoothed as in [15], while ADMM-based methods directly use its subgradient. As for the composite regularizer $r(\mathbf{x})$, all the tested algorithms require computing $\mathrm{M}^{\eta}_{r_k}(\mathbf{x})$ for each group $\mathbf{g}_k$, and the corresponding solution is [23]

$$[\mathrm{P}^{\eta}_{r_k}(\mathbf{x})]_j = \begin{cases} x_j & j \notin \mathbf{g}_k \\ \left[1 - \frac{\eta}{\|\mathbf{x}_{\mathbf{g}_k}\|}\right]_+ x_j & j \in \mathbf{g}_k. \end{cases}$$

Figure 2 shows how the optimization objective varies with the number of iterations. As PA-ASGD also employs smoothing for the hinge loss, there is no noticeable improvement over ANSGD except for $K = 5$. On the other hand, RDA-ADMM is comparable with accelerated stochastic gradient methods, while OPG-ADMM is the slowest.

4.3 Graph-Guided Logistic Regression

In this section, we perform experiments on graph-guided logistic regression [16]:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-l_i \mathbf{x}^T \mathbf{s}_i)) + \lambda \left(\|\mathbf{x}\|^2 + \sum_{\{k_1,k_2\}\in E} |x_{k_1} - x_{k_2}| \right).$$

Here, E is the set of edges for the graph defined on the d variates of $\mathbf{x}$. Following [16], we construct this graph by sparse inverse covariance selection [2]. A similar problem is considered in the generalized lasso [20] and graph-guided SVM [16], though with a different loss function. Note that with the introduction of the $\|\mathbf{x}\|^2$ regularizer, the optimization problem is now strongly convex. For an edge k connecting features k_1 and k_2, $\left[\mathrm{P}^{\eta}_{r_k}(\mathbf{x})\right]_j$ is given by [23]

$$\begin{cases} x_j - \mathrm{sign}(x_{k_1} - x_{k_2}) \min\left\{\eta, \frac{|x_{k_1} - x_{k_2}|}{2}\right\} & j = k_1 \text{ or } k_2; \\ x_j & \text{otherwise.} \end{cases}$$

Experiments are performed on four popular binary classification data sets[2] (Table 1) [18]. For each data set, 80% of the samples are used for training, and the rest for testing. We fix $\lambda = 10^{-4}$, and use 1% of the training samples as mini-batch.

Table 1: Summary of the data sets.

data set	number of samples	dimensionality
a9a	32,561	123
covertype	581,012	54
quantum	50,000	78
sido	12,678	4,932

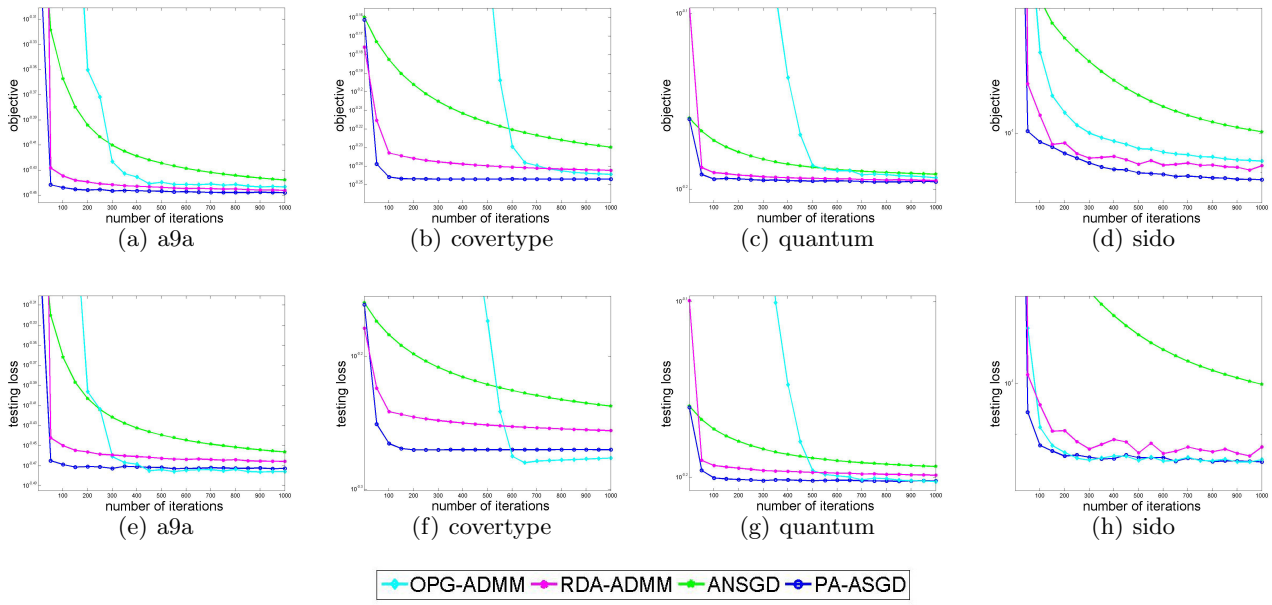
Figure 3 shows the objective values and testing losses obtained by the various algorithms versus the number of iterations. As can be seen, PA-ASGD is the fastest and rapidly leads to a model with good generalization performance. The second best is RDA-ADMM, and ANSGD is the slowest.

4.4 Comparison in the Batch Setting

As discussed in Section 3.3, the proposed method can be used in the batch setting. In this section, we perform experiments on the overlapping group lasso regression:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}(l_i - \mathbf{x}^T\mathbf{s}_i)^2 + \lambda_1\|\mathbf{x}\|^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{x}_{\mathbf{g}_k}\|,$$

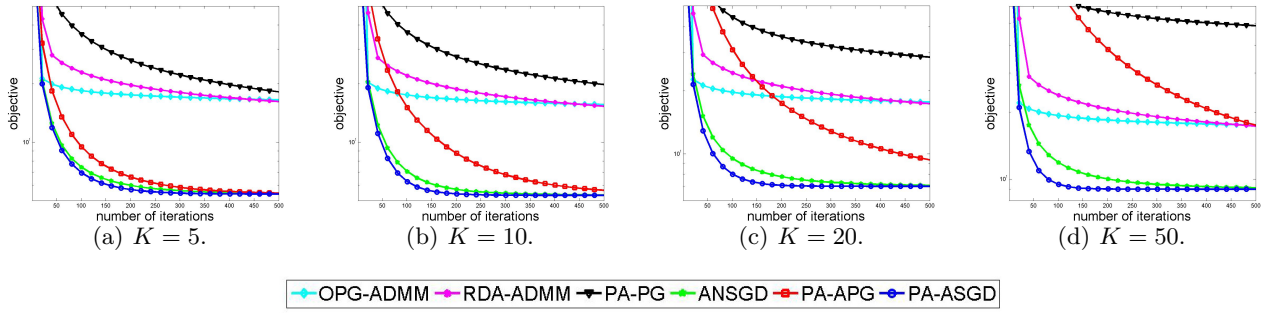
[2]Data sets *a9a* and *covertype* are downloaded from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`, *quantum* is from `http://osmot.cs.cornell.edu/kddcup` and *sido* from `http://www.causality.inf.ethz.ch/home.php`.



(a) a9a (b) covertype (c) quantum (d) sido

(e) a9a (f) covertype (g) quantum (h) sido

Figure 3: Performance versus number of iterations on stochastic graph-guided logistic regression. Top: Objective value; Bottom: Testing loss.



(a) $K = 5$. (b) $K = 10$. (c) $K = 20$. (d) $K = 50$.

Figure 4: Objective value versus number of iterations for batch overlapping group lasso.

where $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-2}$. The data set is generated in the same manner as in Section 4.2, but with a larger $\vartheta_i \sim \mathcal{N}(0, 100)$. In this batch setting, all methods use the whole data set to compute the gradient. Moreover, PA-PG and PA-APG [23], which can only be used in a batch setting, are also included for comparison. Following Theorem 1 of [23], we set its $\mu = 2\epsilon/\bar{L}^2$, where $\bar{L}^2 = \lambda_2^2 K^2$ while $\epsilon = 10^{-2}$ and 10^{-4} for PA-PG and PA-APG, respectively.

Results are shown in Figure 4. As can be seen, the PA-ASGD has the fastest convergence, which is then followed by ANSGD. PA-APG, because of its more conservative stepsize, is worse than the other APG-based methods. The ADMM-based methods and PA-PG are the slowest.

5 Conclusion

In this paper, we developed a novel stochastic accelerated gradient algorithm for regularized risk minimization problems with composite regularizer. Using the proximal average, it enjoys the same computational simplicity as existing stochastic methods based on smoothing or ADMM, but with a faster convergence rate. Empirical results on both general convex and strongly convex problems demonstrate its efficiency over existing methods.

Acknowledgments

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 614311).

References

[1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning*, pages 19–53. 2011.

[2] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[3] A. Barbero and S. Sra. Fast Newton-type methods for total variation regularization. In *Proceedings of the 28th International Conference on Machine Learning*, pages 313–320, June 2011.

[4] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang. The proximal average: Basic theory. *SIAM Journal on Optimization*, 19(2):766–785, 2008.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] S. Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[7] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[8] C. Hu, J. Kwok, and W. Pan. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems 23*, pages 781–789, 2009.

[9] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[10] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems 24*, pages 1558–1566. 2010.

[11] A. Nemirovsky and D. Yudin. Problem complexity and method efficiency in optimization. *Transl. from the Russian by ER Dawson*, 1983.

[12] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Netherlands, 2004.

[13] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[14] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 76, Catholic University of Louvain, 2007.

[15] H. Ouyang and A. Gray. Stochastic smoothing for nonsmooth minimizations: Accelerating SGD by exploiting structure. In *Proceedings of the 29th International Conference on Machine Learning*, July 2012.

[16] H. Ouyang, N. He, L. Tran, and A. Gray. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[17] E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1351–1358, July 2012.

[18] T. Suzuki. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *Proceedings of the 30th International Conference on Machine Learning*, pages 392–400, 2013.

[19] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[20] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

[21] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

[22] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[23] Y. Yu. Better approximation and faster algorithm using the proximal average. In *Advances in Neural Information Processing Systems 26*, 2013.

[24] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, 37(6A):3468–3497, 2009.