Adaptive Proximal Average Based Variance Reducing Stochastic Methods for Optimization with Composite Regularization

Jingchang Liu, Linli Xu, Junliang Guo, Xin Sheng
School of Computer Science and Technology
University of Science and Technology of China, Hefei, China
xdjcl@mail.ustc.edu.cn, linlixu@ustc.edu.cn, {guojunll, xins}@mail.ustc.edu.cn

Abstract

We focus on empirical risk minimization with a composite regulariser, which has been widely applied in various machine learning tasks to introduce important structural information regarding the problem or data. In general, it is challenging to calculate the proximal operator with the composite regulariser. Recently, proximal average (PA) which involves a feasible proximal operator calculation is proposed to approximate composite regularisers. Augmented with the prevailing variance reducing (VR) stochastic methods (e.g. SVRG, SAGA), PA based algorithms would achieve a better performance. However, existing works require a fixed stepsize, which needs to be rather small to ensure that the PA approximation is sufficiently accurate. In the meantime, the smaller stepsize would incur many more iterations for convergence. In this paper, we propose two fast PA based VR stochastic methods – APA-SVRG and APA-SAGA. By initializing the stepsize with a much larger value and adaptively decreasing it, both of the proposed methods are proved to enjoy the $\mathcal{O}(n\log\frac{1}{\epsilon} + m_0\frac{1}{\epsilon})$ iteration complexity to achieve the ϵ-accurate solutions, where m_0 is the initial number of inner iterations and n is the number of samples. Moreover, experimental results demonstrate the superiority of the proposed algorithms.

Introduction

In many artificial intelligence and machine learning applications, one needs to solve the following generic optimization problem in the form of regularized empirical risk minimization (ERM) given n samples (Hastie, Tibshirani, and Friedman 2001):

$$\min_{x\in\mathbb{R}^d} F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + r(x), \qquad (1)$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ denotes the empirical loss of the i-th sample with regard to the parameter x, and r is the regularization term, which is convex but possibly non-smooth. The goal is to find the optimal solution of x that minimizes the regularized empirical loss over the whole dataset.

To solve the problem, deterministic algorithms including traditional gradient descent (GD) and accelerated gradient descent (AGD) (Nesterov 2013) are proposed, which involve the calculation of n gradients. When the data size scales up, n can be rather large, which makes the calculation in deterministic algorithms unaffordable. An effective alternative is stochastic gradient descent (SGD) (Robbins and Monro 1951) which involves lower per iteration cost by utilizing the stochastic gradient instead of the full gradient to update x. However, a rather large variance introduced by the stochastic gradient would slow down the convergence (Bottou, Curtis, and Nocedal 2016). To address this issue, a number of variance reducing (VR) stochastic methods are proposed recently, including SAG (Schmidt, Le Roux, and Bach 2017), SVRG (Johnson and Zhang 2013), SAGA (Defazio, Bach, and Lacoste-Julien 2014) and SARAH (Nguyen et al. 2017). As the key feature of these methods, the variance of stochastic gradient asymptotically goes to zero along the iterative updates, which allows them to achieve the linear convergence rate when each loss is supposed to be strongly convex.

To obtain a compact representation of models, nonsmooth regularisers are often used in regularized ERM problems. Based on the VR stochastic methods mentioned above, a general routine for this case is to employ the forward-backward splitting (Singer and Duchi 2009), which involves the calculation of the proximal operator of r, i.e., $\text{prox}_r^\gamma(\cdot)$. A formal definition of $\text{prox}_r^\gamma(\cdot)$ is

$$\text{prox}_r^\gamma(x) = \operatorname*{arg\,min}_{y\in\mathbb{R}^d}\Big(r(y) + \frac{1}{2\gamma}\left\|y-x\right\|^2\Big), \qquad (2)$$

where $\gamma > 0$. One requirement for using proximal operators is that $\text{prox}_r^\gamma(\cdot)$ can be calculated effectively, such as when $r(x) = \|x\|_1$. But in a large number of important applications in machine learning, such as overlapping group lasso (Jacob, Obozinski, and Vert 2009) and graph-guided fused lasso (Kim and Xing 2009), the regularisers that take a composite form $r(x) = \sum_{k=1}^{K} w_k r_k(x)$ cannot be handled effectively by proximal operators. One feasible technique to fix this issue is the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) as well as its VR stochastic variants, such as SDCA-ADMM (Suzuki 2014) and SVRG-ADMM (Zheng and Kwok 2016). However, in spite of the improved efficiency and scalability, these methods require more space to store the transformation matrix and involve complex implementation and convergence analysis. Very recently, attempts are made to apply three operator splitting to problems with the composite form regularisers (Pedregosa and Gidel 2018; Pedregosa, Fatras, and Casotto 2018), but they make a strong assumption that r is smooth in the analysis of strongly convex case.

An alternative to the above methods is to smooth the nonsmooth regulariser. One traditional way is to utilize Nesterov's smoothing technique (Nesterov 2005; Chen et al. 2012). While recently, Yu (Yu 2013) introduces the proximal average (PA) approximation in accelerated proximal gradient methods (PA-APG), and strictly shows its superiority over the smoothing techniques. Several works have further developed the PA method with different settings. Among them, (Yu et al. 2015) and (Zhong and Kwok 2014) apply the PA approximation to non-convex regularisers. (Zhong and Kwok 2014) combines PA with stochastic gradient methods. Further, PA is introduced to VR stochastic methods by (Cheung and Lou 2017).

On the other hand, approximating a composite regulariser with PA would introduce approximation bias which is proportional to the stepsize according to (Yu 2013), and a small stepsize is required to ensure the accuracy of approximation. In (Yu 2013) and (Cheung and Lou 2017), a fixed stepsize is adopted and set as small as $\mathcal{O}(\epsilon)$, where ϵ is the precision given in advance. This makes the algorithms impractical when a higher-precision solution is required. To tackle this issue, the adaptive stepsize strategy is introduced in (Zhong and Kwok 2014) and (Shen et al. 2017). Instead of setting the stepsize rather small at the beginning, algorithms with adaptive stepsize start with a large stepsize and reduce it gradually. In the two works mentioned above, the stepsize decreases at a rate of $\mathcal{O}(1/k)$, where k is the number of iterations. As a result, these algorithms enjoy significantly fewer iterations than the corresponding fixed stepsize algorithms when one needs a higher-precision solution.

In this paper, we apply the adaptive stepsize strategy to the PA-based VR stochastic algorithms, and propose the following algorithms correspondingly: APA-SVRG and APA-SAGA. Both algorithms consist of two layers of loops, the stepsize iteratively decreases to ρ $(0 < \rho < 1)$ times of the previous stepsize before the inner loop starts. Meanwhile, the number of inner loops inside each outer loop increases accordingly. We prove that the overall number of gradient calculations of the proposed algorithms, APA-SVRG and APA-SAGA, are both $\mathcal{O}(n \log \frac{1}{\epsilon} + m_0 \frac{1}{\epsilon})$ to achieve ϵ-accurate solutions, where m_0 is the initial number of inner iterations. Note that such rate is superior to that of PA-ASGD in (Zhong and Kwok 2014). Compared to PA-based VR stochastic methods with fixed stepsize in (Cheung and Lou 2017), the proposed algorithms need significantly fewer iterations to achieve higher-precision solutions. On the other side, compared to ADMM-based VR stochastic algorithms, the proposed algorithms have low storage requirement as they do not need to store the transformation matrix, and are easier to implement and analyse. The experiments on overlapping group lasso and graph-guided lasso empirically validate the superiority of proposed algorithms.

The rest of this paper is organized as follows. After introducing the problem formulation, assumptions used in the paper and reviewing the relevant theories of VR stochastic methods and proximal average, we respectively present the proposed APA-SVRG and APA-SAGA algorithms, together with the corresponding convergence rate analysis. We then present the experimental results and conclude the paper.

Notation. In this paper, we denote the gradient of the differentiable function f_i at x as $\nabla f_i(x)$. $\|x\|$ and $\|x\|_1$ are the l_2 and l_1 norm of vector x respectively. $\langle \cdot, \cdot \rangle$ denotes the inner product. x^* denotes the point on which F attains its optimal value, which is denoted by F^*. We use x^k for x in the k-th iteration. $\mathbb{E}[\cdot]$ denotes an expected value taken with respect to all choices of indexes up to the current iteration, while $\mathrm{E}[\cdot]$ denotes the expected value taken with respect to the choice of index at the current iteration.

Background and Related Work

In this section, we formulate the problem considered in this paper, together with some common assumptions. Then we overview the theories of variance reducing (VR) stochastic methods and proximal average (PA) which are the foundation of our methods.

Problem Formulation

We consider the following optimization problem

$$\min_{x \in \mathbb{R}^d} F(x) = f(x) + r(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \sum_{k=1}^{K} w_k r_k(x), \tag{3}$$

where $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$. The above formulation defines a general regularized ERM, in which the regulariser r is a convex combination of the K components r_k $(k = 1, 2, \ldots, K)$. Such composite regularisers have shown the superiority in capturing important structural information regarding the problem or data, such as structured sparsity (Zhao, Rocha, and Yu 2009). The following lists the composite forms of r for two representative machine learning models.

- *Overlapping group lasso* (Jacob, Obozinski, and Vert 2009). To select meaningful groups of features, the overlapping group lasso is introduced with the regulariser

$$r(x) = \lambda \sum_{k=1}^{K} \|x_{g_k}\|, \tag{4}$$

where $\lambda > 0$, and g_k indicates the index group of features, and x_{g_k} is the corresponding subvector of x.

- *Graph-guided fused lasso* (Kim and Xing 2009). Graph-guided fused lasso leads to structured sparsity according to the graph $\mathcal{G} \equiv \{\mathcal{V}, \mathcal{E}\}$, in which $\mathcal{V} = \{x_1, \ldots, x_d\}$, where $x_i \in \mathbb{R}$, is the vertex set and $\mathcal{E}$ is the set of edges among $\mathcal{V}$. The regulariser is

$$r(x) = \sum_{\{i,j\} \in \mathcal{E}} w_{ij} |x_i - x_j|, \tag{5}$$

which would penalize the difference among variables connected in $\mathcal{G}$.

To facilitate the analysis, we make the following assumptions on f_i's and r_k's, which are common in optimization. As we focus on problems with smooth loss functions, we assume that each f_i is L-smooth.

Assumption 1. *Each f_i is L-smooth ($L > 0$), namely for any $x, y \in \mathbb{R}^d$,*

$$f_i(y) \leq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{L}{2}\|y - x\|^2. \quad (6)$$

We also assume the strong convexity of f_i. This assumption can be readily satisfied by combing the general convex loss functions with the strongly convex penalties, such as ℓ_2 norm.

Assumption 2. *Each f_i is μ-strongly ($\mu > 0$) convex, namely for any $x, y \in \mathbb{R}^d$,*

$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2. \quad (7)$$

For each non-smooth penalty r_k, we assume it to be Lipschitz continuous with L_k as shown in the following assumption.

Assumption 3. *Each r_k is L_k-Lipschitz continuous, namely for any $x, y \in \mathbb{R}^d$,*

$$|r_k(x) - r_k(y)| \leq L_k\|x - y\|. \quad (8)$$

Variance Reducing Stochastic Methods

The variance of stochastic gradient would limit the performance of stochastic algorithms. To effectively reduce the variance, some methods which employ the control variates (Owen 2013, Chapter 8.9) are introduced in recent years (Johnson and Zhang 2013; Defazio, Bach, and Lacoste-Julien 2014; Defazio, Domke, and others 2014; Schmidt, Le Roux, and Bach 2017). Among them, SVRG (Johnson and Zhang 2013) and SAGA (Defazio, Bach, and Lacoste-Julien 2014) are two representatives, which propose the following VR stochastic gradient:

$$v^k = \nabla f_j(x^k) - \nabla f_j(\tilde{x}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}), \quad (9)$$

where $\tilde{x}$ is the retained 'snapshot' of x to replace the stochastic gradient $\nabla f_j(x^k)$ in SGD. As the control variate of $\nabla f_j(x)$, $\nabla f_j(\tilde{x})$ will asymptotically get closer and closer to $\nabla f_j(x)$ in expectation along the iterative updates, in which case the variance goes to zero. As a result, both SVRG and SAGA can converge under the fixed stepsize, and a large stepsize leads to a much faster convergence rate. Equipped with proximal operators to handle the nonsmooth regularisers, the corresponding algorithms are known as Prox-SVRG (Xiao and Zhang 2014) and Prox-SAGA (Defazio, Bach, and Lacoste-Julien 2014), which involve the following update

$$x^{k+1} = \text{prox}_r^{\gamma}(x^k - \gamma v^k), \quad (10)$$

where $\gamma > 0$ is the fixed stepsize. When the regularization r is simple and admit an efficient proximal operation, Prox-SVRG and Prox-SAGA perform quite well both in practice and theory. Next, we briefly review Prox-SVRG and Prox-SAGA together with the corresponding theories, which would be useful when establishing our own work.

Prox-SVRG consists of two loops. It saves $\tilde{x}$ and calculates the full gradient $\sum_{i=1}^{n} \nabla f_i(\tilde{x})/n$ just before the inner loop begins. $\tilde{x}$ is kept in fixed number of iterations, and updated again just after the current outer loop ends.

The convergence analysis of Prox-SVRG under Assumption 1 and Assumption 2 is established in (Xiao and Zhang 2014). Define

$$\theta = \frac{1}{\gamma\mu(1 - 4L\gamma)m} + \frac{4L\gamma(m+1)}{(1 - 4L\gamma)m}, \quad (11)$$

where γ is the stepsize and m is the number of inner loops inside each outer loop. Further denote $\tilde{x}_s = \sum_{k=1}^{m} x_k/m$ in the s-th outer loop. The change of function value after one outer loop of Prox-SVRG is described as

$$\text{E}F(\tilde{x}_s) - F^* \leq \theta[F(\tilde{x}_{s-1}) - F^*]. \quad (12)$$

Therefore, if $0 < \gamma < 1/(4L)$ and m is large enough such that $\theta < 1$, the linear convergence rate of Prox-SVRG with respect to the outer iterations can be obtained immediately.

Meanwhile in Prox-SAGA, a table is established to record the gradient $\nabla f_i(x_i)$ for each $i = 1, 2, \ldots, n$, here $x_i \in \mathbb{R}^d$ is the value of x at one previous iteration. In this way, at the cost of memory consumption, Prox-SAGA is simpler as it avoids the expensive calculation of the full gradient.

The corresponding theories regarding the convergence of Prox-SAGA under Assumption 1 and Assumption 2 are established in (Defazio, Bach, and Lacoste-Julien 2014). The analysis is based on the Lyapunov function

$$\begin{aligned} T^k = &\frac{1}{n}\sum_{i=1}^{n} f_i(x_i^k) - f(x^*) - \frac{1}{n}\sum_{i=1}^{n} \big\langle \nabla f_i(x^*), x_i^k - x^* \big\rangle \\ &+ c\|x^k - x^*\|^2. \end{aligned} \quad (13)$$

Then the relation between T^k and T^{k+1} can be set up:

$$\begin{aligned} \text{E}[T^{k+1}] - T^k \leq& -\frac{1}{\kappa}T^k + C_1\Big[f(x^k) - f(x^*) - \big\langle \nabla f(x^*), x^k - x^* \big\rangle\Big] \\ &+ C_2\Big[\frac{1}{n}\sum_{i=1}^{n} f_i(x_i^k) - f(x^*) + \frac{1}{n}\sum_{i=1}^{n} \big\langle \nabla f_i(x^*), x_i^k - x^* \big\rangle\Big] \\ &+ C_3 \cdot c\|x^k - x^*\|^2 + C_4 \cdot c\gamma\text{E}\|\nabla f_j(x^k) - \nabla f_j(x^*)\|^2, \end{aligned} \quad (14)$$

where $C_1 = \frac{1}{n} - \frac{2c\gamma(L-\mu)}{L} - 2c\gamma^2\mu\beta$, $C_2 = \frac{1}{\kappa} + 2(1 + \beta^{-1}c\gamma^2 L - \frac{1}{n})$, $C_3 = \frac{1}{\kappa} - \gamma\mu$ and $C_4 = (1 + \beta)\gamma - \frac{1}{L}$. Adopting $\gamma = \frac{1}{3L}$, $c = \frac{1}{2\gamma(1-\gamma\mu)n}$, and $\kappa = \frac{1}{\min\{\frac{1}{4n}, \frac{\mu}{3L}\}}$ together with $\beta = 2$ to ensure that C_1, C_2, C_3 and C_4 are all non-positive, the linear convergence rate of Prox-SAGA can be established since $c\|x^k - x^*\|^2 \leq T^k$:

$$\mathbb{E}\|x^k - x^*\|^2 \leq \Big(1 - \min\big\{\frac{1}{4n}, \frac{\mu}{3L}\big\}\Big)^k T^0. \quad (15)$$

Proximal Average

The proximal operator can readily handle some basic regularisers, such as $r(x) = \|x\|_1$. However, for the composite regularization term $r(x)$ in the form of (3), efficient solutions for $\text{prox}_r^{\gamma}(x)$ are generally hard to obtain although each component $r_k(x)$ can be easily handled. As the approximation to $r(x)$, proximal average (PA) (Bauschke et al. 2008; Yu 2013) is introduced to tackle this issue. Formally, the PA $\hat{r}(x)$ of $r(x)$ is defined below.

Definition 1 (Proximal Average)**.** (Bauschke et al. 2008; Yu 2013) The proximal average of r is the unique semicontinuous convex function $\hat{r}(x)$ such that $M^{\gamma}_{\hat{r}}(x) = \sum_{k=1}^{K} w_k M^{\gamma}_{r_k}(x)$, where $M^{\gamma}_{r}(x) = \inf_y \big(r(y) + \|y - x\|^2 / 2\gamma\big)$.

Since $\nabla M^{\gamma}_{\hat{r}}(x) = \frac{1}{\gamma}(x - \mathrm{prox}^{\gamma}_{\hat{r}}(x))$, the corresponding proximal operator of $\hat{r}(x)$ can be immediately obtained:

$$\mathrm{prox}^{\gamma}_{\hat{r}}(x) = \sum_{k=1}^{K} w_k \cdot \mathrm{prox}^{\gamma}_{r_k}(x). \tag{16}$$

That is to say, we approximate $r(x)$ by pretending the linearity of proximal operators. This approximation may lead to bias, which can be bounded by the following Lemma (Yu 2013).

Lemma 1. *Under Assumption 3, we have* $0 \le r(x) - \hat{r}(x) \le \frac{\gamma \bar{L}^2}{2}$, *where* $\bar{L}^2 = \sum_{k=1}^{K} w_k L_k^2$.

As a result, as the stepsize γ gets smaller, $\hat{r}(x)$ would be closer to $r(x)$. In fact, (Yu 2013) and (Cheung and Lou 2017) adopt the rather small stepsize $\gamma = \mathcal{O}(\epsilon)$ to achieve the desired accuracy ϵ, which would lead to many more iterations when ϵ is small.

Based on the above background and analysis, we develop our methods in the next section to tackle the issues raised by the composite regularization as defined in (3), which cannot be directly solved by traditional VR stochastic methods.

Adaptive Proximal Average based Variance Reducing Stochastic Methods

Although Prox-SVRG and Prox-SAGA perform well when dealing with common problems, they are incapable to handle problems with more complex composite regularisers as defined in (3). A proper alternative is to consider the following approximated problem

$$\min_{x \in \mathbb{R}^d} \hat{F}(x) = f(x) + \hat{r}(x), \tag{17}$$

in which r is replaced by its proximal average $\hat{r}$. Then the iteration (10) becomes

$$x^{k+1} = \mathrm{prox}^{\gamma}_{\hat{r}}(x^k - \gamma v^k), \tag{18}$$

which can be efficiently solved according to the property of proximal average as shown in (16) by

$$x^{k+1} = \sum_{k=1}^{K} w_k \cdot \mathrm{prox}^{\gamma}_{r_k}(x^k - \gamma v^k). \tag{19}$$

And the difference between $F(x)$ and $\hat{F}(x)$ is bounded by $\gamma \bar{L}^2/2$ according to Lemma 1. A straightforward approach to reduce this difference is to adopt a rather small stepsize, as in (Yu 2013; Cheung and Lou 2017), which would lead to a rather slow convergence rate when a high-precision solution is required. A more flexible alternative is to apply the adaptive stepsize (Zhong and Kwok 2014; Shen et al. 2017), which starts with a relatively large stepsize value and gradually decreases it. But for Prox-SVRG and Prox-SAGA specifically, the adjustment of stepsize would influence the convergence, since with the decreasing stepsize we may not ensure θ defined in (11) to be less than 1 and C_1-C_4 defined in (14) to be non-positive. To fix these issues, we develop two adaptive proximal average based algorithms named APA-SVRG and APA-SAGA. Next we will elaborate on these two algorithms with the corresponding convergence analysis respectively.

Adaptive Proximal Average based SVRG

We summarize the idea of adaptive proximal average based SVRG (APA-SVRG) in Algorithm 1. We employ the efficient proximal operator in step 9 by replacing r with $\hat{r}$ as mentioned above. To compensate for the resulting bias, we choose to decrease the stepsize, and increase the number of inner loops inside the outer loop accordingly.

Algorithm 1 APA-SVRG Algorithm

```
 1: Initialize: An initial number of inner loops $m_0 > 0$,
    decay rate $0 < \rho < 1$, and an initial point $\tilde{x}_0$.
 2: for $s = 1, 2, \cdots,$ do
 3:     $x^0 = \tilde{x}_{s-1}$, $\tilde{v} = \sum_{i=1}^{n} f_i(\tilde{x}_{s-1})/n$;
 4:     $m_s = m_0 \cdot \rho^{-s}$;
 5:     $\gamma_s = \min\{1/4L, \rho^s\}$;
 6:     for $l = 1, 2, \cdots, m_s$ do
 7:         Randomly pick $j$ from $\{1, 2, \ldots, n\}$;
 8:         $v^l = \nabla f_j(x^{l-1}) - \nabla f_j(\tilde{x}_{s-1}) + \tilde{v}$;
 9:         $x^l = \sum_{k=1}^{K} w_k \cdot \mathrm{prox}^{\gamma_s}_{r_k}(x^{l-1} - \gamma_s v^l)$;
10:     end for
11:     $\tilde{x}_s = \sum_{l=1}^{m_s} x^l/n$.
12: end for
```

In the following, we denote $\hat{F}_{s+1}$ as the approximation of F with the stepsize parameter γ_{s+1}, and $\hat{F}^*_{s+1}$ as the minimum value of $\hat{F}_{s+1}$. According to Lemma 1, we have

$$F(\tilde{x}_s) - F^* \le \hat{F}_s(\tilde{x}_s) - F^* + \frac{\gamma_s}{2}\bar{L}^2. \tag{20}$$

In general, $F(\tilde{x}_s) - F^*$ decreases with a linear rate as we have shown in (12). To keep this linear convergence rate for $\hat{F}_s(\tilde{x}_s) - F^*$, we set the decay rate of γ_s to be linear as shown in step 5 to balance $\hat{F}_s(\tilde{x}_s) - F^*$ and $\gamma_s \bar{L}^2/2$. Also, γ_s need to be less than $1/(4L)$ to ensure a positive θ defined in (11). Moreover, we increase the number of inner loops from the initial m_0 exponentially as shown in step 4, to ensure that there exists $0 < \theta < 1$, such that

$$\mathrm{E}\hat{F}_{s+1}(\tilde{x}_{s+1}) - \hat{F}^*_{s+1} \le \theta(\hat{F}_{s+1}(\tilde{x}_s) - \hat{F}^*_{s+1}). \tag{21}$$

We state our main theorem for APA-SVRG followed with its proof below.

Theorem 1 (APA-SVRG)**.** *Suppose that Assumption 1, 2 and 3 hold. Then for the update in APA-SVRG, it holds that*

$$\begin{aligned} &\mathbb{E}F(\tilde{x}_s) - F^* \\ &\le \theta^s\big(\hat{F}_0(\tilde{x}_0) - F^*\big) + \frac{\gamma_0}{2}\bar{L}^2 \frac{\theta}{\theta - \rho}\big(\theta^s - \rho^s\big). \end{aligned}$$

Proof. According to Lemma 1, we have

$$F(\tilde{x}_{s+1}) - F^* \leq \hat{F}_{s+1}(\tilde{x}_{s+1}) - F^* + \frac{\gamma_{s+1}}{2}\bar{L}^2. \quad (22)$$

Now, we bound the expectation of $\hat{F}_{s+1}(\tilde{x}_{s+1}) - F^*$ by (21):

$$\begin{aligned}
&\mathbb{E}\hat{F}_{s+1}(\tilde{x}_{s+1}) - F^* \\
&= \mathbb{E}\big[\hat{F}_{s+1}(\tilde{x}_{s+1}) - \hat{F}^*_{s+1} + \hat{F}^*_{s+1} - F^*\big] \\
&\leq \mathbb{E}\big[\theta\big(\hat{F}_{s+1}(\tilde{x}_s) - \hat{F}^*_{s+1}\big) + \hat{F}^*_{s+1} - F^*\big] \\
&= \mathbb{E}\big[\theta\big(\hat{F}_s(\tilde{x}_s) - F^*\big) + \big(1-\theta\big)\big(\hat{F}^*_{s+1} - F^*\big) \\
&\quad + \theta\big(\hat{F}_{s+1}(\tilde{x}_s) - \hat{F}_s(\tilde{x}_s)\big)\big], \qquad (23)
\end{aligned}$$

where $0 < \theta < 1$. As $F^* = F(x^*) \geq \hat{F}_{s+1}(x^*) \geq \hat{F}^*_{s+1}$, we have

$$\hat{F}^*_{s+1} - F^* \leq 0. \quad (24)$$

Meanwhile, denote $\hat{r}_s$ and $\hat{r}_{s+1}$ as the approximation of r with stepsize γ_s and γ_{s+1} respectively, given

$$\begin{aligned}
&\hat{F}_{s+1}(\tilde{x}_s) - \hat{F}_s(\tilde{x}_s) = \hat{r}_{s+1}(\tilde{x}_s) - \hat{r}_s(\tilde{x}_s) \\
&= r(\tilde{x}_s) - \hat{r}_s(\tilde{x}_s) + \hat{r}_{s+1}(\tilde{x}_s) - r(\tilde{x}_s)
\end{aligned}$$

and $0 \leq r(\tilde{x}_s) - \hat{r}_s(\tilde{x}_s) \leq \gamma_s\bar{L}^2/2$, $-\gamma_{s+1}\bar{L}^2/2 \leq \hat{r}_{s+1}(\tilde{x}_s) - r(\tilde{x}_s) \leq 0$ from Lemma 1, we have

$$-\frac{\gamma_{s+1}}{2}\bar{L}^2 \leq \hat{F}_{s+1}(\tilde{x}_s) - \hat{F}_s(\tilde{x}_s) \leq \frac{\gamma_s}{2}\bar{L}^2. \quad (25)$$

Plugging (24) and (25) into (23), we get

$$\mathbb{E}\hat{F}_{s+1}(\tilde{x}_{s+1}) - F^* \leq \theta \cdot \mathbb{E}\big(\hat{F}_s(\tilde{x}_s) - F^*\big) + \theta\frac{\gamma_s}{2}\bar{L}^2. \quad (26)$$

Summing up the above inequality over $0, 1, \ldots, s$ yields

$$\begin{aligned}
&\mathbb{E}\hat{F}_{s+1}(\tilde{x}_{s+1}) - F^* \\
&\leq \theta^{s+1}\big(\hat{F}_0(\tilde{x}_0) - F^*\big) + \big(\theta^{s+1}\frac{\gamma_0}{2}\bar{L}^2 + \cdots + \theta\frac{\gamma_s}{2}\bar{L}^2\big).
\end{aligned}$$

Plugging this inequality into (22) with taking expectation on each term, we have

$$\begin{aligned}
&\mathbb{E}F(\tilde{x}_{s+1}) - F^* \\
&\leq \theta^{s+1}\big(\hat{F}_0(\tilde{x}_0) - F^*\big) + \big(\theta^{s+1}\frac{\gamma_0}{2} + \cdots + \theta^0\frac{\gamma_{s+1}}{2}\big)\bar{L}^2 \\
&\leq \theta^{s+1}\big(\hat{F}_0(\tilde{x}_0) - F^*\big) + \frac{\gamma_0}{2}\bar{L}^2\frac{\theta}{\theta-\rho}\big(\theta^{s+1} - \rho^{s+1}\big),
\end{aligned}$$

where the second inequality holds due to $\gamma_s \leq \rho^s$. □

Apparently, when $\rho = 1$, which means the stepsize is fixed, $\mathbb{E}F(\tilde{x}_{s+1})$ will not converge to the minimum value, and when $0 < \rho < 1$, $F(\tilde{x}_{s+1}) - F^*$ approaches 0 at the exponential rate. So to achieve the ϵ-accurate solution, the total number of outer loops denoted as S is $\mathcal{O}(\log\frac{1}{\epsilon})$. Then we have the following corollary about the total number of required gradient calculations.

Corollary 1. *To achieve the ϵ-accurate solution, the overall iteration complexity of APA-SVRG is $\sum_{s=0}^{S}\mathcal{O}(n + 2m_s) = \mathcal{O}(nS + \sum_{s=0}^{S} m_s) = \mathcal{O}(n\log\frac{1}{\epsilon} + m_0\frac{1}{\epsilon})$.*

Note that the rate $\mathcal{O}(n\log\frac{1}{\epsilon} + m_0\frac{1}{\epsilon})$ is faster than that deduced by Theorem 2 in (Zhong and Kwok 2014).

Adaptive Proximal Average based SAGA

Next, the algorithm of adaptive proximal average based SAGA (APA-SAGA) is summarized in Algorithm 2. Unlike traditional Prox-SAGA (Defazio, Bach, and Lacoste-Julien 2014) which contains only one layer of loops, APA-SAGA utilizes a multi-stage scheme to progressively decease the stepsize as in APA-SVRG.

Algorithm 2 APA-SAGA Algorithm

```
 1: Initialize: An initial number of inner loops $m_0 > 0$, decay rate $0 < \rho < 1$, an initial point $x^0$, and $g_i^0 = \nabla f(x^0), i = 1, 2, \ldots, n$.
 2: for $s = 1, 2, \cdots,$ do
 3:     $m = m_0 \cdot \rho^{-s}$;
 4:     $\gamma_s = \frac{1}{3L} \cdot \rho^s$;
 5:     $x^0 = x_s$;
 6:     for $l = 1, \cdots, m$ do
 7:         Randomly pick $j$ from $\{1, 2, \ldots,$ n$\}$;
 8:         $v^l = \nabla f_j(x^{l-1}) - g_j^l + \sum_{i=1}^n g_i^l/n$;
 9:         $x^l = \sum_{k=1}^K w_k \cdot \mathrm{prox}_{r_k}^{\gamma_s}(x^{l-1} - \gamma_s v^l)$;
10:         Update $g_i^l,\ i = 1, 2, \ldots, n$:
                $g_i^l = \begin{cases} \nabla f_j(x^{l-1}), & \text{if } i = j, \\ g_i^{l-1}, & \text{otherwise.} \end{cases}$
11:     end for
12:     $x_s = x^m$.
13: end for
```

At time k in the s-th outer loop, we define the Lyapunov function T_s^k as

$$\begin{aligned}
T_s^k \;=\; &\frac{1}{n}\sum_{i=1}^n f_i(x_i^k) - f(\hat{x}^*) - \frac{1}{n}\sum_{i=1}^n \big\langle \nabla f_i(\hat{x}^*), x_i^k - \hat{x}^* \big\rangle \\
&+ c\|x^k - \hat{x}^*\|^2, \qquad (27)
\end{aligned}$$

where $\hat{x}^*$ is the minimum of the approximated function $\hat{F}$. The same as (14), we have

$$\begin{aligned}
\mathrm{E}[T_s^{k+1}] - T_s^k \leq &-\frac{1}{\kappa}T_s^k + C_1\Big[f(x^k) - f(\hat{x}^*) - \big\langle \nabla f(\hat{x}^*), x^k - \hat{x}^* \big\rangle\Big] \\
&+ C_2\Big[\frac{1}{n}\sum_{i=1}^n f_i(x_i^k) - f(\hat{x}^*) + \frac{1}{n}\sum_{i=1}^n \big\langle \nabla f_i(\hat{x}^*), x_i^k - \hat{x}^* \big\rangle\Big] \\
&+ C_3 \cdot c\|x^k - \hat{x}^*\|^2 + C_4 \cdot c\gamma\mathrm{E}\|\nabla f_j(x^k) - \nabla f_j(\hat{x}^*)\|^2, \qquad (28)
\end{aligned}$$

where C_1, C_2, C_3 and C_4 are defined in (14). Since the decreasing stepsize is required, we adopt $\gamma = \frac{1}{3L}\rho^s$ here, as shown in step 4, together with $c = \frac{3L}{2(1-\mu\gamma)n}$, $\kappa = \frac{1}{\min\{\frac{1}{4n}, \frac{\mu}{3L}\rho^{-s}\}}$, $\beta = 2\rho^{-s}$ to ensure that C_1, C_2, C_3 and C_4 are non-positive. Then chaining over m yields

$$\mathbb{E}T_s^m \leq \Big(1 - \min\big\{\frac{1}{4n}, \frac{\mu}{3L}\rho^s\big\}\Big)^m T_s^0. \quad (29)$$

According to the Bernoulli's Inequality, it holds that

$$1 \geq \big(1 - \min\{\frac{1}{4n}, \frac{\mu}{3L}\rho^s\}\big)^m \geq 1 - m \cdot \min\{\frac{1}{4n}, \frac{\mu}{3L}\rho^s\}. \quad (30)$$

Since ρ^s $(0 < \rho \leq 1)$ decays with the increase of s in the exponential rate, we increase the number of inner loops exponentially as well, as shown in step 3 of APA-SAGA, to ensure that there exists $0 < \theta < 1$ such that $\mathrm{E}T_s^m \leq \theta \cdot T_s^0$.

Before the formal theorem regarding the convergence rate, we propose a lemma which establishes the relation between T_{s+1}^0 and T_s^m. Due to space limitation, the proof details are put into supplementary materials.

Lemma 2. *Suppose that Assumptions 1, 2 and 3 hold and for the iterate set $\{x^k\}_{k=0,1,2,\ldots}$ produced in APA-SAGA, the radius defined by*

$$R := \sup_{k=0,1,2,\ldots} \|x^k - x^*\|$$

is bounded, that is, $R < +\infty$.[1] *Then the following inequality holds*

$$T_{s+1}^0 \leq T_s^m + \rho^{s/2} \cdot D_1 + \rho^s \cdot D_2, \tag{31}$$

where $D_1 = 2RL\big(1 + \frac{9L}{(3L-\mu)n}\big)\sqrt{\frac{\bar{L}^2}{\mu}}$, $D_2 = 4L\big(1 + \frac{9L}{2(3L-\mu)n}\big)\frac{\bar{L}^2}{\mu}$.

Based on this lemma, we have the following theorem for APA-SAGA.

Theorem 2 (APA-SAGA). *Suppose that Assumptions 1, 2 and 3 hold. Then for the update in APA-SAGA, it holds that*

$$\begin{aligned}
&\mathbb{E}\|x_s - x^*\|^2 \\
\leq\ & \frac{4n}{3L}T_0^0 \cdot \theta^{s+1} + \frac{\bar{L}^2}{\mu}\frac{2}{3L}\rho^s + \theta\frac{\theta^s - \rho^{s/2}}{\theta - \rho^{1/2}} \cdot \frac{4n}{3L}D_1 \\
&+ \theta\frac{\theta^s - \rho^s}{\theta - \rho} \cdot \frac{4n}{3L}D_2.
\end{aligned}$$

Similar to APA-SVRG, with the fixed stepsize which corresponds to $\rho = 1$, we cannot ensure the convergence of $\mathbb{E}\|x_s - x^*\|^2$ to 0. On the other hand, when $0 < \rho < 1$, we can deduce the following corollary which is analogous to Corollary 1.

Corollary 2. *To achieve the ϵ-accurate solution, the overall iteration complexity of APA-SAGA is $\mathcal{O}(n \log \frac{1}{\epsilon} + m_0 \frac{1}{\epsilon})$.*

Experiments

In this section, we conduct experiments on overlapping group lasso and graph-guide fused lasso to verify the effectiveness of our proposed APA-SVRG and APA-SAGA algorithms.

Experimental Setup

We compare the following methods in our experiments.

- The proposed APA-SVRG and APA-SAGA.
- PA-SVRG and PA-SAGA (Cheung and Lou 2017): proximal average based methods, which need a rather small stepsize when a higher-precision solution is desired.
- SVRG-ADMM (Zheng and Kwok 2016): stochastic ADMM combined with variance reduction.
- PA-ASGD (Zhong and Kwok 2014): accelerated stochastic gradient descent with proximal average.

Since the Nesterov's smoothing based algorithms, e.g. ANSGD (Ouyang and Gray 2012) as well as the deterministic gradient descent methods, e.g. PA-PG (Yu 2013), have been shown to be inferior to PA-ASGD in (Zhong and Kwok 2014), we do not compare with them. Moreover, we choose SVRG-ADMM without the accelerated technique in the experiment for a fair comparison. To establish the formulation for ADMM, we refer to (Qin and Goldfarb 2012) for the overlapping group lasso problem and (Ouyang et al. 2013) for the graph-guided fused lasso problem.

We use synthetic datasets in the overlapping group lasso experiment and four real datasets from LIBSVM (Chang and Lin 2011) in the graph-guided fused lasso experiment. The real datasets are summarized in Table 1. We tune the stepsize and other parameters for different algorithms so that they can achieve the best performance, such as in APA-SAGA and APA-SVRG, ρ is set to about 0.8 to enable a proper decay rate. To make a fair comparison, the initial value of x is set to zero for all algorithms. Denote the number of samples as n, we measure the *objective value* at x as $F(x)$ and the *number of iterations* as the evaluation of n component gradients to evaluate the performance of algorithms.

Overlapping Group Lasso

We first conduct experiments on the overlapping group lasso model (Jacob, Obozinski, and Vert 2009) with the squared loss:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}(x^\intercal a_i - b_i)^2 + \lambda\sum_{k=1}^{K}\|x_{g_k}\|, \tag{32}$$

where $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ and $\lambda > 0$. Similar to (Yu 2013), all entries in a_j $(j = 1, 2, \ldots, n)$ are sampled from *i.i.d.* normal distribution, $x_j = (-1)^j \exp(-(j-1)/100)$, $b_j = x^\intercal a_j + \xi$ with the noise $\xi \sim \mathcal{N}(0, 1)$, and the groups are defined as

$$\underbrace{\{1, \ldots, 100\}}_{g_1}, \underbrace{\{91, \ldots, 190\}}_{g_2}, \ldots, \underbrace{\{d - 99, \ldots, d\}}_{g_K},$$

where $d = 90K + 10$. We set $\lambda = K/(5n)$ and vary K in $\{5, 10, 20, 50\}$. Moreover, we set $n = d$. For the composite regulariser $r_k(x)$, $\mathrm{prox}_{r_k}^{\gamma}(x)$ for each group g_k can be readily computed as

$$(\mathrm{prox}_{r_k}^{\gamma}(x))_j = \begin{cases} x_j & j \notin g_k; \\ \big(1 - \frac{\gamma}{\|x_{g_k}\|}\big)_+ x_j & j \in g_k. \end{cases}$$

For PA-SVRG and PA-SAGA, to explore the effect of accuracy on the convergence, we set the desired accuracy $\epsilon = 10^{-4}$ when $K = 5$ and $K = 10$, $\epsilon = 10^{-5}$ when $K = 50$. And when $K = 20$, we set $\epsilon = 10^{-4}$ for PA-SVRG, $\epsilon = 10^{-5}$ for PA-SAGA. The experimental results are shown in Figure 1. As can be seen, PA-SVRG and PA-SAGA need rather small stepsize when the desired accuracy is high, which greatly inhibits their performance. On

[1] This assumption has also appeared in some existing works, e.g. (Liu et al. 2015).

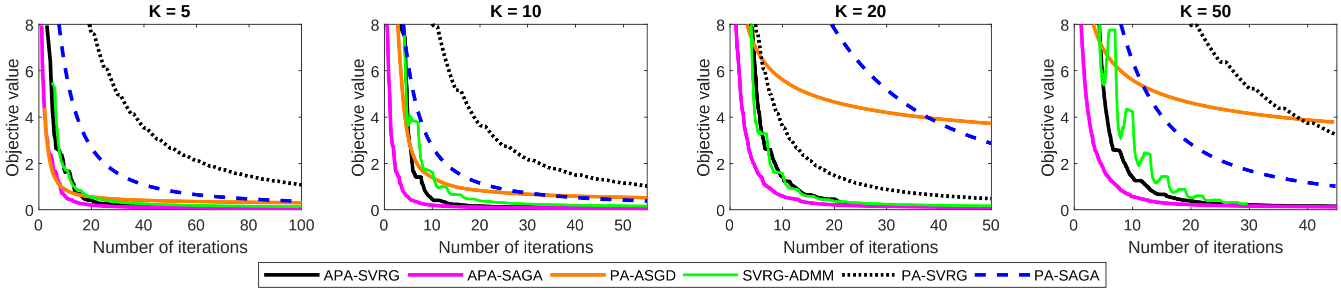

Figure 1: Comparison of several algorithms with overlapping group lasso.

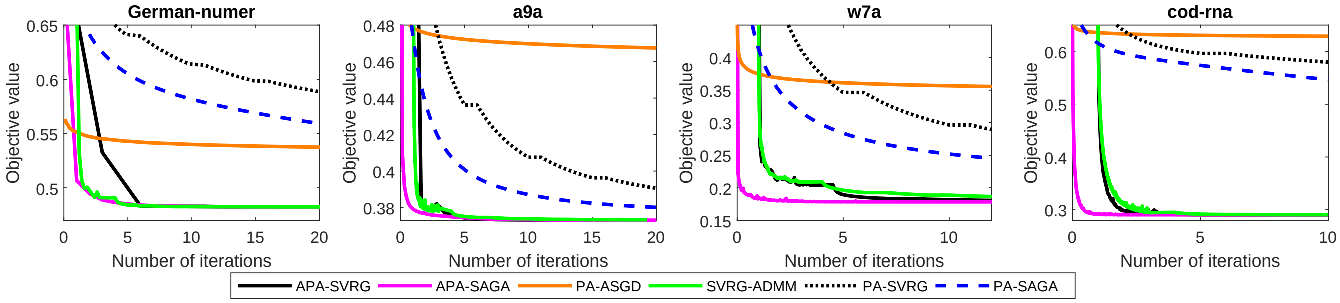

Figure 2: Comparison of several algorithms with graph-guided fused lasso.

Table 1: Summary of the datasets used in the graph-guided fused lasso experiments

Dataset	#samples	dimensionality	λ
german-numer	1000	24	10^{-3}
a9a	32561	123	10^{-4}
w7a	24692	300	10^{-4}
cod-rna	59353	8	10^{-4}

the other hand, the adaptive stepzise strategy enables APA-SAGA and APA-SVRG to achieve a fast convergence rate while involving a simpler implementation as well as convergence analysis than SVRG-ADMM. Moreover, the performance of PA-ASGD is poor on all datasets.

Graph-Guided Logistic Regression

We proceed with the experiments on the graph-guided logistic regression (Kim and Xing 2009):

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-b_i x^\intercal a_i)) \\ +\lambda\big(\|x\|^2+\sum_{\{k_1,k_2\}\in E}|x_{k_1}-x_{k_2}|\big). \qquad (33)$$

Here, E is the graph edge set, and we construct this graph by sparse inverse covariance selection (Friedman, Hastie, and Tibshirani 2008). For an edge k connecting feature k_1 and k_2, $\mathrm{prox}_{r_k}^{\gamma}(x)$ can be easily computed, and the value on its j-th index is

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

We set the desired accuracy $\epsilon = 10^{-4}$ for PA-SVRG and PA-SAGA. Figure 2 shows the experimental results. As can be seen, the performance of PA-ASGD is poor without the variance reduction technique. On the other side, the small stepsize limits the performance of PA-SVRG and PA-SAGA. With a larger stepsize, APA-SVRG, APA-SAGA and SVRG-ADMM can quickly approach the optimal point. And the adaptive stepsize strategy enables the iterations of APA-SVRG and APA-SAGA to converge to the optimal point.

Conclusion

In this paper, we apply the adaptive stepsize strategy to the PA-based VR stochastic algorithms, and propose the corresponding APA-SVRG and APA-SAGA algorithms. By initializing the stepsize with a relatively large value and adaptively decreasing it, both proposed algorithms can achieve the $\mathcal{O}(n\log\frac{1}{\epsilon}+m_0\frac{1}{\epsilon})$ iteration complexity. Moreover, experiments on overlapping group lasso and graph-guided logistic regression demonstrate the efficiency of the proposed methods.

Acknowledgement

Research supported by the National Natural Science Foundation of China (No. 61673364, No. 91746301) and the Fundamental Research Funds for the Central Universities (WK2150110008).

References

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

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2016. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.; et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3(1):1–122.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, X.; Lin, Q.; Kim, S.; Carbonell, J. G.; Xing, E. P.; et al. 2012. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics* 6(2):719–752.

Cheung, Y.-m., and Lou, J. 2017. Proximal average approximated incremental gradient descent for composite penalty regularized empirical risk minimization. *Machine Learning* 106(4):595–622.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 1646–1654.

Defazio, A.; Domke, J.; et al. 2014. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning*, 1125–1133.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3):432–441.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Jacob, L.; Obozinski, G.; and Vert, J.-P. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, 433–440. ACM.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, 315–323.

Kim, S., and Xing, E. P. 2009. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS genetics* 5(8):e1000587.

Liu, J.; Wright, S. J.; Ré, C.; Bittorf, V.; and Sridhar, S. 2015. An asynchronous parallel stochastic coordinate descent algorithm. *The Journal of Machine Learning Research* 16(1):285–322.

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

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

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, volume 70, 2613–2621. PMLR.

Ouyang, H., and Gray, A. 2012. Stochastic smoothing for nonsmooth minimizations: Accelerating sgd by exploiting structure. *arXiv preprint arXiv:1205.4481*.

Ouyang, H.; He, N.; Tran, L.; and Gray, A. 2013. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, 80–88.

Owen, A. B. 2013. *Monte Carlo theory, methods and examples*.

Pedregosa, F., and Gidel, G. 2018. Adaptive three operator splitting. In Dy, J., and Krause, A., eds., *International Conference on Machine Learning*, 4085–4094.

Pedregosa, F.; Fatras, K.; and Casotto, M. 2018. Variance reduced three operator splitting. *arXiv preprint arXiv:1806.07294*.

Qin, Z., and Goldfarb, D. 2012. Structured sparsity via alternating direction methods. *Journal of Machine Learning Research* 13(May):1435–1468.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics* 400–407.

Schmidt, M.; Le Roux, N.; and Bach, F. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162(1-2):83–112.

Shen, L.; Liu, W.; Huang, J.; Jiang, Y.-G.; and Ma, S. 2017. Adaptive proximal average approximation for composite convex minimization. In *AAAI*, 2513–2519.

Singer, Y., and Duchi, J. C. 2009. Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems 22*, 495–503.

Suzuki, T. 2014. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *International Conference on Machine Learning*, 736–744.

Xiao, L., and Zhang, T. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* 24(4):2057–2075.

Yu, Y.; Zheng, X.; Marchetti-Bowick, M.; and Xing, E. 2015. Minimizing nonconvex non-separable functions. In *Artificial Intelligence and Statistics*, 1107–1115.

Yu, Y.-L. 2013. Better approximation and faster algorithm using the proximal average. In *Advances in neural information processing systems*, 458–466.

Zhao, P.; Rocha, G.; and Yu, B. 2009. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics* 3468–3497.

Zheng, S., and Kwok, J. T. 2016. Fast-and-light stochastic admm. In *IJCAI*, 2407–2613.

Zhong, W., and Kwok, J. 2014. Accelerated stochastic gradient method for composite regularization. In *Artificial Intelligence and Statistics*, 1086–1094.