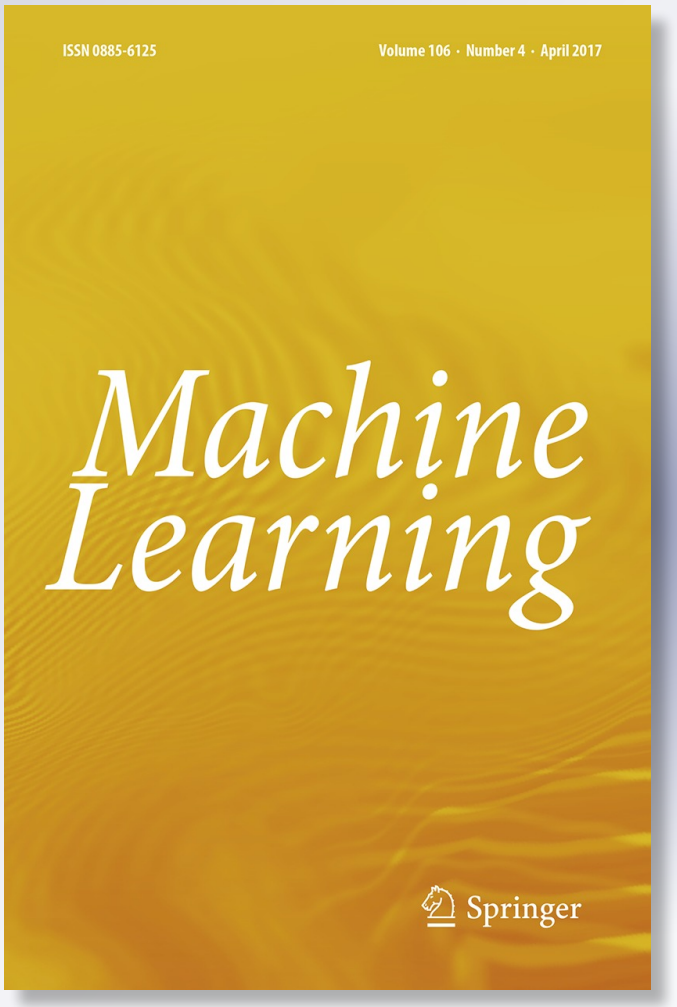
Proximal average approximated incremental gradient descent for composite penalty regularized empirical risk minimization

Yiu-ming Cheung & Jian Lou

Machine Learning

ISSN 0885-6125
Volume 106
Number 4

Mach Learn (2017) 106:595-622
DOI 10.1007/s10994-016-5609-1



Springer

Mach Learn (2017) 106:595–622
DOI 10.1007/s10994-016-5609-1

Proximal average approximated incremental gradient descent for composite penalty regularized empirical risk minimization

Yiu-ming Cheung[1] · **Jian Lou**[1]

Received: 16 February 2016 / Accepted: 7 November 2016 / Published online: 18 November 2016

Abstract Composite penalties have been widely used for inducing structured properties in the empirical risk minimization (ERM) framework in machine learning. Such composite regularizers, despite their superior performance in grasping structural sparsity properties, are often nonsmooth and even nonconvex, which makes the problem difficult to optimize. Proximal average (PA) is a recently proposed approximation technique targeting these regularizers, which features the tractability of implementation and theoretical analysis. However, current PA-based methods, notwithstanding the promising performance of handling composite penalties against traditional techniques, are either slow in convergence or do not scale well to large datasets. To make PA an ideal technique for optimizing ERM with composite penalties, this paper proposes a new PA-based algorithm called IncrePA by incorporating PA approximation into an incremental gradient framework. The proposed method is a more optimal PA-based method that features lower per-iteration cost, a faster convergence rate for convex composite penalties, and guaranteed convergence for even nonconvex composite penalties. Experiments on both synthetic and real datasets demonstrate the efficacy of the proposed method in optimizing convex and nonconvex ERM with composite penalties.

Keywords Empirical risk minimization · Composite regularizer · Proximal average · Incremental gradient descent

Editors: Geoff Holmes, Tie-Yan Liu, Hang Li, Irwin King and Zhi-Hua Zhou.

✉ Yiu-ming Cheung
ymc@comp.hkbu.edu.hk

Jian Lou
jianlou@comp.hkbu.edu.hk

[1] Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong, SAR, China

1 Introduction

Empirical risk minimization (ERM) is a fundamental machine learning method that learns a model by minimizing the average loss taken from the training data. To induce better prediction performance and introduce prior knowledge about the model, the empirical loss is often regularized by a penalty function. Based on the specific task, the penalty functions can vary from smooth functions like squared ℓ_2-norm to nonsmooth simple functions like ℓ_1-norm. Composite nonsmooth functions, featuring their capability of inducing a structured sparsity model, have been intensively utilized in bioinformatics and text mining tasks. However, it is difficult to efficiently optimize such composite penalty regularized ERM problems, especially when confronted with very large datasets.

In general, nonsmooth composite penalties, like overlapping group lasso or graph-guided lasso, are hard to deal with. One fact is that the proximal gradient method (Beck and Teboulle 2009; Nesterov and Nesterov 2004), which is an effective approach to simple nonsmooth penalties, is not applied in this case because its crucial proximal mapping step is difficult to solve. That is, existing simple methods cannot be directly applied when engaging with these complex structured penalties. A splitting method called alternating direction method of multipliers (ADMM) (Boyd et al. 2011), with its variants like stochastic ADMM and incremental ADMM with better scalability, has been extensively studied. Stochastic ADMM methods (Ouyang et al. 2013) utilize stochastic gradient updating strategies to reduce per-iteration computational cost. For example, RDA-ADMM (Suzuki 2013) incorporates the RDA method with ADMM; SADMM and optimal-SADMM in Azadi and Sra (2014) utilize nonuniform averaging of the iterative variable (Lacoste-Julien et al. 2012; Shamir and Zhang 2013) and accelerated stochastic gradient method (Ghadimi and Lan 2012) to further accelerate the stochastic ADMM method. Incremental ADMM methods (Zhong and Kwok 2014a; Suzuki 2014) can achieve a faster convergence rate than stochastic ADMM by utilizing the incremental gradient updating strategy. In particular, SA-ADMM (Zhong and Kwok 2014a) and SDCA-ADMM (Suzuki 2014) are two recently proposed ADMM methods incorporating two different incremental gradient methods: SAG (Roux et al. 2012) and SDCA (Shalev-Shwartz and Zhang 2013) respectively. However, despite the above effort for better efficiency and scalability, a remaining major drawback of ADMM-based methods is the complex implementation and convergence analysis, which are brought about by the additional variables introduced and the alternating updating scheme.

Recently, an alternative to ADMM called proximal average (PA) (Yu 2013) has been introduced to efficiently handle composite penalties. It approximates the original composite penalty when each constituent regularizer admits a simple proximal map. The resulting proximal average approximation then enjoys a simple proximal map by averaging the proximal map of its components. What makes the PA technique interesting is that the approximation can be controlled arbitrarily close to the original composite regularizer and be strictly better than the smoothing technique. Compared with ADMM, Zhong and Kwok (2014c) points out that ADMM is also a proximal method by duplicating variables. As will be seen later, proximal average is not only simple for implementation, but is also easier for theoretical analysis. Along this line, pioneering work includes the one in Yu (2013), which introduces proximal average with the accelerated full gradient method FISTA (Beck and Teboulle 2009). Zhong and Kwok (2014c) incorporates the proximal average technique with the stochastic variant of the optimal gradient method. It has provable superiority over the smoothing technique which is also shared by Yu (2013). Despite the simplicity advantage in terms of implementation and analysis, when compared to incremental ADMM methods (e.g. SA-ADMM and

SDCA-ADMM), existing PA-based approaches either converge slowly (e.g. PA-ASGD) or suffer from high per-iteration cost (e.g. PA-APG).

Incremental gradient methods featuring both scalability and the fast convergence property have been receiving considerable attention as an efficient approach to mitigating the ever growing dataset problem. As these methods only calculate gradients associated with a randomly picked data sample in each iteration as stochastic gradient methods (Bottou 2010; Xiao 2010; Ghadimi and Lan 2012), they have comparable low per-iteration computation cost. More importantly, by exploiting the finite sum structure of the loss function which stochastic methods do not have, these incremental methods are able to achieve a linear convergence rate as per full gradient methods (Nesterov and Nesterov 2004). For example, SAG (Roux et al. 2012) utilizes the average of the stored past gradients, one for each data sample. SVRG (Johnson and Zhang 2013; Xiao and Zhang 2014) adopts a multi-stage scheme to progressively control the variance of the stochastic gradient. Both methods have linear convergence rate for strongly convex problems, but the theoretical convergence result for general convex loss is still unclear. SAGA (Defazio et al. 2014a) has both a sublinear convergence guarantee for general convex loss and linear convergence for strongly convex loss. It is a midpoint of SAG and SVRG by taking both update patterns from them in its iteration. There are also other incremental methods like FINITO (Defazio et al. 2014b) and MISO (Mairal 2014), which consume more memory because they not only store the gradient, but also the variable. S2GD (Konečnỳ and Richtárik 2013) is a method very similar to SVRG with the difference only in stage length. SDCA (Shalev-Shwartz and Zhang 2013) is a dual incremental method.

The above-mentioned methods mainly focus on convex composite penalties. Nonconvex composite penalties, although leading to an even more difficult problem, can have better prediction performance by avoiding the over-penalization problems of their convex counterparts. For structured sparsity inducing tasks, there has been some research incorporating structured sparsity regularizers with nonconvex penalties and showing improved prediction performance (Shen and Huang 2010; Xiang et al. 2013). For optimizing such nonconvex composite penalties, general nonconvex solvers like the concave-convex procedure (CCCP) (Zhang 2010) and the sequential convex program (SCP) (Lu 2012) proceed in a multi-stage convexify scheme that solves a convex relaxation in each stage up to a certain approximation and then constructs a convex surrogate for the next stage. Zhong and Kwok (2014b) has recently proposed a proximal average based gradient descent method called GD-PAN for such a penalty. It has been shown that it is still possible to approximate the nonconvex composite function with proximal average for some common nonconvex penalties. Also, by solving such a surrogate, it is more efficient than multi-stage methods like CCCP and SCP, because the proximal map of the proximal gradient descent can be easily computed for the surrogate. However, GD-PAN that is essentially a batch gradient method suffers from the scalability problem. In this paper, we also propose an incremental proximal average method for solving nonconvex composite penalty problems.

Here, we investigate the potential to incorporate incremental gradient methods with the proximal average technique. For convex composite penalties, we show that, by solving a surrogate problem, the proposed method can achieve linear convergence when the loss function is strongly convex and sublinear convergence when the loss is general convex. By contrast, ADMM-based methods cannot provide both. For example, SDCA-ADMM only has convergence results for strongly convex loss, while the convergence analysis of SAG-ADMM only applies when the loss is general convex. Furthermore, we also extend the incremental PA technique to solve nonconvex penalty problems, which has better scalability than the

batch method GD-PAN (Zhong and Kwok 2014b). In this setting, we show that the proposed method converges to an asymptotic stationary point of the surrogate problem.

The remainder of this paper is organized as follows: Sect. 2 introduces the notation and assumptions used in this paper. Section 3 conducts an overview of PA and incremental gradient descent methods. In Sect. 4, we propose our method for convex composite penalties with strongly convex loss and general convex loss, and establish the corresponding convergence rate. Section 5 proposes an incremental proximal average algorithm for solving nonconvex composite penalty problems. Section 6 shows the experimental results for both convex composite penalty problems and nonconvex composite penalty problems on synthetic and real datasets. Finally, Sect. 7 concludes the paper.

2 Preliminaries

In this section, we firstly introduce the notation used in this paper. Then, we formally define the problem to be optimized. Also, we will describe the assumptions for these problems.

Notation In the following, we denote the gradients of the differentiable function l_i and l at x as $\nabla l_i(x)$ and $\nabla l(x)$, respectively. $||x||_2$ and $||x||_1$ denote the l_2 and l_1 norm of vector x correspondingly. $\langle \nabla l_i(x), y\rangle$ is the inner product of $\nabla l_i(x)$ and y. The superscript $(\cdot)^T$ stands for the transpose of $(\cdot)$. We denote the t-th iteration of x by x^t. We assume the dataset is indexed as $1, 2, \ldots, n$, and the subscript i like x_i is related to the i-th data sample. We denote the k-th component of the composite penalty function by the subscript k in r_k.

We consider the following ERM with composite penalty problem:

$$\min_{x\in\mathbb{R}^d} F(x) = l(x) + r(x) = \frac{1}{n}\sum_{i=1}^{n} l_i(x) + \sum_{k=1}^{K} \alpha_k r_k(x), \tag{1}$$

$$\sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0, \tag{2}$$

which is commonly applied to learn the model defined by variable x from training data set $\{\xi_i, y_i\}$ $i = 1, \ldots, n$. ξ_i is the data vector, and y_i is its label. In (1), $l_i(x)$ is the loss taken at data sample (ξ_i, y_i) with index i. The function $r(x)$ is the composite penalty for regularization purpose, which is composed by K constituent regularizers. We hide the constant balancing the loss and the regularizer in the loss as Yu (2013) and Zhong and Kwok (2014c), so that $r(x)$ is a convex combination of the K components $r_k(x)$. In this paper, we allow both $l(x)$ and $r(x)$ to be either convex or nonconvex.

Smooth loss function We assume $l_i(x)$ to be smooth with L Lipschitz continuous gradient, so that we can take the gradient for gradient descent and also we are able to construct a local majorization surrogate. Formally, an L-smooth loss function l_i satisfies the following inequality,

Assumption 1 The loss function is L-smooth, $\forall x, y$,

$$l_i(y) - l_i(x) - \langle \nabla l_i(x), y - x\rangle \leq \frac{L}{2}||y - x||_2^2. \tag{3}$$

If we further assume $l_i(x)$ is general convex, l_i also satisfies the following inequality:

Assumption 2 $l_i(x)$ is convex if $\forall x, y$,

$$l_i(y) - l_i(x) - \langle \nabla l_i(x), y - x \rangle \geq 0. \tag{4}$$

Examples of the general convex smooth loss functions include least square loss, logistic loss, and smooth hinge loss, all of which will be used in Sect. 6. In addition, $l_i(x)$ can be strongly convex provided that the following assumption holds:

Assumption 3 $l_i(x)$ is strongly convex if there is a $\mu > 0$ such that $\forall x, y$,

$$l_i(y) - l_i(x) - \langle \nabla l_i(x), y - x \rangle \geq \frac{\mu}{2}||y - x||_2^2. \tag{5}$$

For example, when combining the above general convex loss with a large margin inducing penalty $\frac{\lambda}{2}||x||^2$, it becomes a λ-strongly convex loss.

Composite penalty We focus on composite penalty in this paper, i.e. $r(x)$ is an average of K simple non-smooth penalties $r_k(x)$. We assume that r_k is Lipschitz continuous with the constant M_k, i.e.

Assumption 4 r_k is M_k Lipschitz continuous, $\forall x, y$,

$$|r_k(x) - r_k(y)| \leq M_{r_k}||x - y||_2. \tag{6}$$

Also, the proximal update step of each r_k should be simple. Please note that the proximal map of $r(x)$ itself can be very complex and computationally expensive. In addition, we introduce the notation related to proximal step:

$$M_{r_k}^{\eta}(x) = \min_y \frac{1}{2\eta}||x - y||_2^2 + r_k(y), \tag{7}$$

and

$$P_{r_k}^{\eta}(x) = \arg\min_y \frac{1}{2\eta}||x - y||_2^2 + r_k(y). \tag{8}$$

3 Overview of PA and incremental gradient descent methods

This section gives an overview of the PA technique and the incremental gradient framework.

3.1 Proximal average

Proximal average (Bauschke et al. 2008; Yu 2013) has been recently introduced to deal with composite regularizers. It admits a compact calculation when each single component satisfies Assumption 4. PA only requires each component of $r(x)$ has simple proximal map, even when it is computationally expensive for $r(x)$ itself. The following definition describes the PA $\hat{r}(x)$ of $r(x)$.

Definition 1 [PA (Bauschke et al. 2008; Yu 2013)] The PA of r is the unique semicontinuous convex function $\hat{r}(x)$ such that $M_{\hat{r}(x)}^{\eta} = \sum_{k=1}^{K} \alpha_k M_{r_k}^{\eta}$. The corresponding proximal map of the PA $\hat{r}(x)$ is

$$P_{\hat{r}}^{\eta}(x) = \sum_{k=1}^{K} \alpha_k P_{r_k}^{\eta}(x). \tag{9}$$

Therefore, once approximating $r(x)$ by $\hat{r}(x)$, we can obtain the proximal map of $\hat{r}(x)$ by simply averaging the proximal map of each constituent regularizer $r_k(x)$. The next lemma shows that the approximation of $\hat{r}(x)$ can be controlled arbitrarily close to $r(x)$ by the step size η.

Lemma 1 (Yu 2013) *Under Assumption 4, we have* $0 \leq r(x) - \hat{r}(x) \leq \frac{\eta \bar{M}^2}{2}$, *where* $\bar{M}^2 = \sum_{k=1}^{K} \alpha_k M_k^2$.

In fact, although Yu (2013) verifies the above lemma provided that $r_k(x)$ is convex, GD-PAN shows that it actually applies to nonconvex cases as long as Assumption 2 holds.

3.2 Incremental gradient descent methods

The incremental gradient methods proposed recently make an improvement on stochastic gradient methods provided that the training data is finite. Generally, at each iteration, these methods approximate the full gradient by a combination of a random gradient evaluated at the latest variable with past gradients. There are several types of incremental gradient method. For example, SAG utilizes a gradient table to record past gradients for each data sample index. SVRG uses a single full gradient evaluated periodically. Both of the methods have linear convergence for strongly convex and smooth problems. SAGA shares part of the update pattern from both SAG and SVRG, and has theoretical guarantees for both general convex and strongly convex problems.

Denote the variable table at iteration t by ϕ^t, which contains n vectors recording the iterate x^t in a randomly-select-and-replace strategy. That is, the algorithm randomly selects an index i^t from 1 to N and then replaces the i^{t+1}-th column of ϕ^t by the latest iterate x^t, i.e.

$$\phi_i^{t+1} = \begin{cases} \nabla x^t, & i = i^t \ (Replace) \\ \nabla \phi_i^t, & i \neq i^t \ (Unchanged). \end{cases} \tag{10}$$

Let $\nabla l_i(\phi_i^t)$ $(i = 1, 2, \ldots, n)$ be the gradient table. SAGA, like SAG, updates the random i^t-th gradient with $\nabla l_{i^t}(x^t)$ while keeping other terms unchanged:

$$\nabla l_i(\phi_i^{t+1}) = \begin{cases} \nabla l_i(x^t), & i = i^t \ (Replace) \\ \nabla l_i(\phi_i^t), & i \neq i \ (Unchanged). \end{cases} \tag{11}$$

Hence, we only need to evaluate the gradient related to the i^t data sample by computing $\nabla l_{i^t}(x^t)$. Also, the variable table ϕ^t is introduced for notational convenience and thus need not be explicitly stored.

Based on the stored gradient table, SAG proposes to construct a variance reduced gradient estimation by averaging the gradient table, i.e. $G^t = \frac{1}{n}\sum_{i=i}^{n} \nabla l_i(\phi_i^t)$. On the contrary, SVRG proposes to use the unbiased estimation $G^t = \nabla l_{i^t}(x^t) - \nabla l_{i^t}(\tilde{x}^s) + \frac{1}{n}\sum_{i=1}^{n} \nabla l_i(\tilde{x^s})$, where $\frac{1}{n}\sum_{i=1}^{n} \nabla l_i(\tilde{x}^s)$ is the batch gradient evaluated periodically on $\tilde{x}^s$ (e.g. every $2N$ iterations). SAGA propose to approximate the gradient for iteration t:

$$G^t = \nabla l_{i^t}(\phi_i^{t+1}) - \nabla l_{i^t}(\phi_i^t) + \frac{1}{n}\sum_{i=1}^{n} \nabla l_i(\phi_i^t). \tag{12}$$

SAGA shows that this gradient estimation strategy actually stands in middle of that used by SAG and SVRG. Also, conditioned on information up to the t-th iteration, G^t is an unbiased estimation of the full gradient in expectation. According to Johnson and Zhang (2013), Xiao

and Zhang (2014), such approximate gradients have the reduced variance, which would lead to speed up over stochastic methods. SAGA admits iteration schemes involving proximal mapping, but only for simple penalty functions equipping closed-form update and is incapable to handle the more complex composite penalties.

4 Accelerated proximal average approximated incremental gradient for ERM with convex composite penalty

In this section, we present the proposed incremental gradient descent proximal average method for convex composite penalty regularized ERM problems, which is termed as IncrePA-cvx. We first illustrate the convex composite penalty functions with two types of structured sparsity inducing penalties as examples, i.e. overlap group lasso and graph-guided lasso. We then describe the proposed method provided with the convergence rate for convex composite penalties with general convex and strongly convex loss.

4.1 Overlapping group lasso and graph-guided fused lasso

In the following, we describe two convex composite regularizers for inducing structured sparsity among features in sparsity estimation tasks.

Overlapping group lasso Jacob et al. (2009) introduces overlapping group lasso

$$r(x) = \sum_{k=1}^{K} \omega_k r_k(x) = \sum_{k=1}^{K} \frac{1}{K} ||x_{g_k}||_2, \tag{13}$$

where g_k indicates the index group of features, and x_{g_k} is a copy of x with the values of those that are not in the index subset g_k being set at 0. Apparently, the proximal map of each individual $||x_{g_k}||$ is simple to compute, while the proximal map of $r(x)$ is difficult due to the coupled nature of overlapping groups of indices.

Graph-guided fused lasso Kim and Xing (2009) induces structured sparsity according to the graph $\mathscr{E}$,

$$r(x) = \sum_{k=1}^{K} \omega_k r_k(x) = \sum_{k=1}^{K} \omega_k |x_{k_1} - x_{k_2}|, \tag{14}$$

where $\{k_1, k_2\} \in \mathscr{E}$. Again, the proximal map of $r(x)$ is not easy to compute even though $r_k(x)$ is.

4.2 Incremental gradient proximal average for convex composite penalty regularized ERM

The proposed method proceeds with a proximal gradient style iterative scheme. With the estimated gradient utilized in iteration t denoted by G^t and step size by η, the algorithm updates:

$$x_{t+1} = \arg\min_x \frac{1}{2\eta} ||x - (x_t - \eta G^t)||_2^2 + r(x), \tag{15}$$

which can be denoted by $x^{t+1} = P_r^{\eta}(x^t - \eta G^t)$ [recall that the proximal map corresponding to penalty function $r(x)$ is denoted by $P_r^{\eta}(\cdot)$, as shown in (8)]. The gradient G^t is estimated

by incremental gradient strategy, in particular SAGA (Defazio et al. 2014a), which consumes low per-iteration cost and fast convergence by reducing the variance of the estimated gradient.

In general, the proximal map corresponding to composite penalties $r(x)$ is not easy to compute. Popular approaches propose to deal with composite penalty functions based on the splitting method ADMM. When coupled with incremental gradient estimation strategies, ADMM-based methods are difficult to analyze. For example, SA-ADMM (based on SAG and linearized ADMM) only has convergence results when the loss function is general convex, while SDCA-ADMM only has convergence results when the loss function is locally strongly convex. Most recently, SVRG-ADMM (Zheng and Kwok 2016) is able to provide the convergence analysis for both general and strongly convex losses, but they require different iteration design under the different convexity assumptions. Hence, to develop a general algorithm that is capable to cover both general and strongly loss function cases with a unified iteration design scheme, we propose to approximate $r(x)$ with proximal average approximation $\hat{r}(x)$. The iteration becomes

$$x^{t+1} = P^{\eta}_{\hat{r}}(x^t - \eta G^t), \tag{16}$$

which can be simply computed according to the proximal average property as shown in (9) by

$$x^{t+1} = \sum_{k=1}^{K} \alpha_k P^{\eta}_{r_k}(x^t - \eta G^t), \tag{17}$$

where $P^{\eta}_{r_k}(\cdot)$ is the proximal mapping with respect to simple constituent function r_k. By utilizing the proximal average update technique, we are actually solving the surrogate problem $\hat{f}(x) = l(x) + \hat{r}(x)$, which can be controlled arbitrarily close to the original problem $F(x)$ according to Lemma 1. We summarize the closeness property by the following lemma.

Lemma 2 *For $r(x)$ with M_k-Lipschitz continuous $r_k(x)(k = 1, \ldots, K)$ and denote $\bar{M}^2 = \sum_{k=1}^{K} \alpha_k M_k^2$ as in Lemma 1, we have $F(x) - \hat{f}(x) \leq \epsilon$ for any x, when we set $\eta \leq \frac{2\epsilon}{\bar{M}^2}$.*

The proposed incremental gradient PA for ERM with convex composite penalty is summarized in Algorithm 1.

Algorithm 1 IncrePA-cvx

Input: η (step size); x_0 (initial value); $\nabla l_i(\phi_i^0), \phi_i^0 = x_0, i = 1, \ldots, n$ (initial table of gradients).
1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Randomly pick $i^t \in \{1, 2, \ldots, n\}$;
3: Update the derivative table as in (11);
4: Calculate G^t by (12);
5: $w^{t+1} = x^t - \eta G^t$;
6: $x^{t+1} = \sum_{k=1}^{K} \alpha_k P^{\eta}_{r_k}(w^{t+1})$;
7: **end for**

4.3 Analysis of IncrePA-cvx

The proposed method is general in the sense that it is provided with convergence analysis covering both general convex loss and strongly convex loss functions cases with a unified iteration design. We describe them as follows.

A. Convergence analysis for general convex objectives In this paragraph, we establish the convergence rate of IncrePA when applied to general convex objectives. Recall the notation of the surrogate function $\hat{f}(x) = l(x) + \hat{r}(x)$ implicitly solved by IncrePA. The following theorem summarizes the sublinear convergence rate:

Theorem 1 *Under Assumption 1 (i.e. l_i is smooth) with $l_i(x)$ general convex and Assumption 4 (i.e. r_k is simple and Lipschitz continuous), let $\hat{x}^*$ be the optimal point of the surrogate problem. Denote Q^t as*

$$Q^t = \frac{1}{n}\sum_{i=1}^{n} l_i(\phi_i^t) - l(\hat{x}^*) - \frac{1}{n}\sum_{i=1}^{n}\langle \nabla l_i(\hat{x}^*), \phi_i^t - \hat{x}^*\rangle. \tag{18}$$

Then, after $t \geq \frac{1}{c_2\epsilon}\left(Q^0 + \left(c_1 + \frac{c_2}{\eta}\right)||x^0 - \hat{x}^*||_2^2\right)$ iterations, we have

$$\mathbb{E}\left[F(\bar{x}^t) - F(\hat{x}^*)\right] \leq 2\epsilon, \tag{19}$$

where $\bar{x}^t = \frac{1}{t}\sum_{i=1}^{t} x^i$. In addition, possible choices of the parameters c_1, c_2, η appearing in the proof are as follows: $\eta < \min\left(\frac{1}{2L}, \frac{2\epsilon}{M^2}\right)$, $c_1 = \frac{1}{2\eta n}$, $c_2 = \frac{1}{2n}\left(\frac{1}{2\eta L\beta} - 1\right)$.

B. Convergence analysis for strongly convex objectives If we further have the strong convexity of the loss function, the proposed method can achieve linear convergence as shown in Theorem 2.

Theorem 2 *Under Assumption 1 (i.e. l_i is smooth) with $l_i(x)$ μ-strongly convex and Assumption 4 (i.e. r_k is simple and Lipschitz continuous), let $\hat{x}^*$ be the optimal point of the surrogate problem. Denote a Lyapunov function T^t as:*

$$T^t = Q^t + \left(c_1 + \frac{c_2}{\eta}\right)||x^t - \hat{x}^*||_2^2 + c_2\left(\hat{f}(x^t) - \hat{f}(\hat{x}^*)\right), \tag{20}$$

$$Q^t = \frac{1}{n}\sum_{i=1}^{n} l_i(\phi_i^t) - l(\hat{x}^*) - \frac{1}{n}\sum_{i=1}^{n}\langle \nabla l_i(\hat{x}^*), \phi_i^t - \hat{x}^*\rangle, \tag{21}$$

where t is the iteration number. After $(1 - \frac{1}{\kappa})(\log\frac{T^0}{\epsilon})$ iterations, we then have

$$\mathbb{E}\left[F(x^t) - F(\hat{x}^*)\right] \leq 2\epsilon. \tag{22}$$

In addition, there exists some $\beta \geq 1$ and possible choices of the parameters c_1, c_2, κ, η appearing in the proof are as follows: $\eta < \min\left(\frac{1}{2L}, \frac{2\epsilon}{M^2}, \frac{1}{2n\mu}\right)$, $c_1 = \frac{1}{2\eta n}\frac{L}{L-\mu}$, $c_2 = c_1\eta\left(\frac{1}{2\eta L\beta} - 1\right)$, $\frac{1}{\kappa} = \frac{2\eta\mu}{1+\frac{1}{2\eta L}}$.

The detailed mathematical proofs of Theorems 1 and 2 are given in "Appendix".

4.4 Discussion

We have the following three remarks to make about the above two convergence guarantees.

Remark 1 First we point out the specialties of the step size parameter η. One can find that we represent all parameters by the step size η in the above convergence analysis because it controls the approximation by Lemma 1. The convergence rate of the strongly convex case is related to $\frac{1}{\kappa}$, i.e. it converges faster when $\frac{1}{\kappa}$ is larger, which depends on L, μ and ϵ given

the data size n. Please note that, for an ill-conditioned problem where $\frac{L}{u} = n$, $\frac{1}{2n\mu}$ can be converted to $\frac{1}{2L}$. Thus, the convergence speed is related to L and ϵ. The convergence speed for the general convex case depends on c_2, i.e. the larger c_2, the faster it converges. Given the dataset size, the convergence speed is again related to L and ϵ.

Remark 2 Like the other incremental methods, the above convergence only reflects training loss (Suzuki 2014; Zhong and Kwok 2014a; Roux et al. 2012). The generalization performance is unknown partly because of the assumption of the finite training set size. Our experiments on testing loss show empirical results of the generalization performance.

Remark 3 Furthermore, our algorithm will converge to the optimal point of the surrogate function. We show the convergence rate by measuring the loss with respect to the objective function value at $\hat{x}^*$ ($F(\hat{x}^*)$), which is different from the usual convention that measures $F(x^*)$. Nevertheless, the surrogate problem will be close to the original problem given a sufficiently small step size, which is therefore able to provide satisfactory generalization performance. Indeed, the experimental results in Sect. 6 have verified that, as a good approximation to the original problem, the proposed method has satisfactory generalization performance in terms of classification error and test loss on the test sets of two real datasets.

5 Incremental proximal average for nonconvex composite penalty regularized ERM

In this section, we extend the incremental gradient with proximal average algorithm to nonconvex composite penalty regularized ERM problems. We first describe example nonconvex composite penalties utilized in structured sparsity estimation tasks, which replace the convex ℓ_1 norm with tighter nonconvex surrogate functions of the ℓ_0 norm. After recalling existing approaches for this type of problem, we present a more scalable method by extending the IncrePA-cvx in the previous section to nonconvex composite penalty case, termed IncrePA-ncvx.

5.1 Two examples of nonconvex composite penalties in structured sparse estimation

Nonconvex composite penalties appear in nonconvex structured sparsity estimation applications. The nonconvex surrogate penalties like capped ℓ_1 norm, smoothly clipped absolute deviation (SCAD) and minimax concave penalty (MCP), are able to address the bias of the convex ℓ_1 norm, thus are considered better relaxations of the ℓ_0 norm for promoting sparsity. Inspired by this, Shen and Huang (2010), Xiang et al. (2013), Zhong and Kwok (2014b) have proposed nonconvex structured sparsity inducing counterparts by wrapping the convex composite functions with nonconvex functions. That is, the penalty function $r(x)$ takes the composite form as an average of K nonconvex composite penalties as

$$r(x) = \sum_{k=1}^{K} \omega_k r_k(x). \tag{23}$$

In this nonconvex composite penalty case, each r_k takes the following form:

$$r_k(x) = \rho(h_k(x)), \tag{24}$$

where $\rho(\cdot)$ is the nonconvex sparsity-inducing function.

In this structured case, compared with traditional non-structured nonconvex relaxations of lasso, it is wrapped outside each constituent convex regularizer rather than each of the indices of x. We elaborate (23) with Capped-ℓ_1 overlapping group-lasso and MCP graph-guided fused Lasso as two concrete examples.

Capped-ℓ_1 overlapping group-lasso This is a hybrid nonconvex composite penalty of Capped-ℓ_1 norm and overlapped group-lasso, which wraps each of the group indices $h_k(x) = ||x_{g_k}||_2$ with Capped-ℓ_1 norm (Zhang 2010):

$$r(x) = \sum_{k=1}^{K} \omega_k r_k(x) = \sum_{k=1}^{K} \omega_k \rho(||x_{g_k}||_2) = \sum_{k=1}^{K} \omega_k \min\{||x_{g_k}||_2, \theta\}, \tag{25}$$

where θ is a constant defining the ℓ_1 norm.

MCP graph-guided fused lasso This nonconvex composite penalty combines MCP (Zhang 2010) with graph-guided fused lasso.

$$r(x) = \sum_{k=1}^{K} \omega_k r_k(x) = \sum_{k=1}^{K} \omega_k \rho(|x_{k_1} - x_{k_2}|), \tag{26}$$

where $\{k_1, k_2\} \in \mathscr{E}$, $|\mathscr{E}| = K$ and $\rho(\cdot)$ takes the following form based on MCP norm:

$$\rho(u) = \begin{cases} \lambda|u| - \frac{u^2}{2a}, & |u| \le a\lambda, \\ \frac{a\lambda^2}{2}, & |u| > a\lambda, \end{cases} \tag{27}$$

where λ and a are constants.

5.2 Related work

Such composite form and nonconvexity make the problem even more difficult to solve. Some existing approaches are proposed with inefficiency or scalability issues. DC programming-based methods like the concave-convex procedure (CCCP) (Zhang 2010) progress by stages that solve a convex surrogate in each stage by approximating nonconvex $r(x)$ with a convex function. This multistage style can be inefficient. General iterative shrinkage and thresholding (GIST) (Gong et al. 2013) and sequential convex program (SCP) (Lu 2012) can be efficient for regularizers with simple proximal update. However, since the proximal step is very difficult for (23), these methods are also not efficient enough. Recently, GD-PAN (Zhong and Kwok 2014b) has extended proximal average for nonconvex (23) and approximates r with proximal average in the GIST algorithm to obtain a proximal update efficient algorithm. However, GD-PAN is intrinsically a batch gradient algorithm with poor scalability towards large-scale problems. Apparently, a PA-based method with better scalability is more attractive and useful from a practical perspective.

5.3 Nonconvex extension of incremental gradient with PA

We aim to extend the incremental gradient PA method to solve these nonconvex structured problems, termed IncrePA-ncvx. Zhong and Kwok (2014b) also approximates the nonconvex composite regularizer with PA, and then solves the approximate problem based on Gong et al. (2013) iteration scheme, which is a batch gradient method. Our method improves upon Gong et al. (2013) with an incremental gradient strategy that results in better scalability.

In this nonconvex case, we also approximate $r(x)$ with its PA approximation $\hat{r}(x)$, which is similar to convex case. For convenience, we denote the PA approximated objective as

$$\arg\min_x \hat{f}(x) = \arg\min_x \frac{1}{n}\sum_{i=1}^{n} \hat{f}_i(x) = \arg\min_x \frac{1}{n}\sum_{i=1}^{n}[l_i(x) + \hat{r}(x)], \tag{28}$$

where each component function $\hat{f}_i(x) = l_i(x) + \hat{r}(x)$ corresponds to the i-th data sample. The PA approximated function $\hat{f}(x)$ is not guaranteed to be convex. Hence, directly applying incremental proximal gradient decent method to $\hat{f}_i(x)$ can hardly ensure convergence. In this regard, we further approximate $\hat{f}(x)$ iteratively with the first-order surrogate of $\hat{f}(x)$ by following Mairal (2014), which is a particular majorization by taking the smoothness of $l_i(x)$ into consideration. Again, as an incremental method, we keep a variable table and a gradient table, in which we denote them again by ϕ_i^t and $\nabla l_i(\phi_i^t)$ correspondingly for the i-th sample at iteration t, by the random choose-and-replace strategy as in the previous section. At iteration t, with the latest variable table and gradient table, a majorization approximation $g_i^t(x)$ of $\hat{f}_i^t(x)$ is constructed as

$$g_i^t(x) = l_i(\phi_i^t) + \langle \nabla l_i(\phi_i^t), x - \phi_i^t\rangle + \frac{1}{2\eta}||x - \phi_i^t||_2^2 + \hat{r}(x), \tag{29}$$

where η is the step size and satisfies $\frac{1}{\eta} \geq L$. By the smoothness assumption of the loss function $l_i(x)$ [assumption (3)], function $g_i^t(x)$ upper bounds $\hat{f}_i(x)$ [i.e. $g_i^t(x) \geq \hat{f}_i(x)$]. Then, in each iteration, the majorization function is minimized with

$$\begin{aligned} x^{t+1} &= \arg\min_x \bar{g}^t(x) = \arg\min_x \frac{1}{n}\sum_{i=1}^{n} g_i^t(x) \\ &= \arg\min_x \frac{1}{n}\sum_{i=1}^{n}\left[l_i(\phi_i^t) + \langle \nabla l_i(\phi_i^t), x - \phi_i^t\rangle + \frac{1}{2\eta}||x - \phi_i^t||_2^2 + \hat{r}(x)\right], \end{aligned} \tag{30}$$

which is an incremental majorization-minimization iteration by choosing the majorization function as the so-called first-order surrogate (Mairal 2014). With such surrogates during iteration, we need extra memory to explicitly store the variable table as compared with convex incremental gradient PA method, where the variable table is introduced only for notational convenience and need not be kept. However, this overhead in memory seems indispensable, because the per-iteration problem evaluated in the previous section cannot be guaranteed to be a majorization of $\hat{f}(x)$, which is obvious when we rewrite the iterate scheme of Algorithm 1 in the same style as (30),

$$x^{t+1} = \arg\min_x \frac{1}{n}\sum_{i=1}^{n}\left[l_i(x_t) + \langle \nabla l_i(\phi_i^t), x - x^t\rangle + \frac{1}{2\eta}||x - x^t||_2^2 + \hat{r}(x)\right]. \tag{31}$$

Then, (30) can be further simplified to $x^{t+1} = \arg\min_x \frac{1}{2\eta}||x - \left(\frac{1}{n}\sum_{i=1}^{n}\phi_i^t - \eta G^t\right)||_2^2 + \hat{r}(x)$, where $G^t = \frac{1}{n}\sum \nabla l_i(\phi_i^t)$. By the property of the PA approximation function $\hat{r}(x)$ and the proximal mapping notation as in (9), we then have

$$x^{t+1} = \sum_{k=1}^{K} \alpha_k P_{r_k}^{\eta}\left(\frac{1}{n}\sum_{i=1}^{n}\phi_i^t - \eta G^t\right). \tag{32}$$

We summarize the above iteration scheme IncrePA-ncvx as shown in Algorithm 2.

Algorithm 2 IncrePA-ncvx

Input: η (step size); x^0 (initial variable); $\nabla l_i(\phi_i^0), i = 1, \ldots, n$ (initial table of gradients); $\phi_i^0, i = 1, \ldots, n$ (initial table of iterate x).
1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Randomly pick $i^t \in \{1, 2, \ldots, n\}$;
3: Update the derivative table as in (11);
4: Update the variable table as in (10);
5: Calculate G^t by averaging the gradient table;
6: $w^{t+1} = \frac{1}{N}\sum_{i=1}^{N} \phi_i^t - \eta G^t$;
7: $x^{t+1} = \sum_{k=1}^{K} \alpha_k P_{r_k}^{\eta}(w^{t+1})$;
8: **end for**

5.4 Analysis of IncrePA-ncvx

The main per-iteration computational cost comes from: (i) step 5 evaluates a stochastic gradient and (ii) step 7 computes the proximal mapping with respect to K simple regularizers and takes the average. Hence, compared to the PA-based method GD-PAN, the proposed method provides better scalability when the dataset size grows because the per-iteration computational cost does not depend on the number of data points.

For nonconvex problems, it is generally impossible to guarantee a global optimum or derive a convergence rate like those for convex and strongly convex problems. Following Mairal (2014), we only provide the convergence of IncrePA-ncvx in the sense that the PA approximation $[\hat{f}(x^t)]$ is almost sure convergence and the sequence $[x^t]$ satisfies the so-called asymptotic stationary point condition (for more details, see Borwein and Lewis 2010).

Definition 2 Asymptotic stationary point: Denote the directional derivative of function f at x^t as $\nabla f(x^t, x - x^t)$ (see Subsection 2.1 in Borwein and Lewis 2010 for the detailed definition), under the assumption that f is bounded below and for all x, x^t, the directional derivative $\nabla f(x, x - x^t)$ of f at x^t in the direction $x - x^t$ exists, the sequence $[x^t]_{t=1,2,\ldots}$ satisfies the asymptotic stationary point condition if

$$\lim_{k\to} \inf_{+\infty} \inf_{x\in\mathcal{X}} \frac{\nabla f(x^t, x - x^t)}{||x - x^t||_2} \geq 0. \tag{33}$$

We rely on the convergence result from Mairal (2014), through which we have the following lemma:

Lemma 3 *Suppose $f(x) = \sum_{i=1}^{n} f_i(x)$ is bounded below and the directional derivative exists. With $g_{i'}^t(x)$ being first-order surrogates and incremental majorization-minimization scheme, $f(x^t)$ is almost sure convergence and x^t satisfies the asymptotic stationary point condition with probability one.*

Based on Lemma 3, we have the convergence result for our IncrePA-ncvx as summarized in the following Theorem 3.

Theorem 3 *Algorithm IncrePA-ncvx is almost sure convergence and the iterates x^t converges to the asymptotic stationary point of the surrogate problem $\hat{f}(x)$ with probability one.*

Proof To utilize Lemma 3, we first observe that our surrogate function $g_i^t(x)$ in (29) is the so-called first-order surrogate of the PA approximation function $\hat{f}_i^t(x)$ in (28). Namely, (i) $g_i^t(x)$ majorizes $\hat{f}_i^t(x)$, i.e. $g_i^t(x) \geq \hat{f}_i^t(x)$; (ii) Denote the approximation error by $h_i^t(x) =$

Table 1 Summary of four real datasets

Data set	Data points	Dimensionality
20 newsgroup	12, 995	100
a9a	32, 561	123
Covtype	581, 012	54
Protein	145, 751	74

$g_i^t(x) - \hat{f}_i^t(x) = l_i^t(\phi_i^t) - l(x) + \langle \nabla l_i^t(\phi_i^t), x - \phi_i^t \rangle + \frac{1}{2\eta}||x - \phi_i^t||_2^2$, then $h_i^t(x)$ is smooth and $h_i^t(\phi_i^t) = 0, \nabla h_i^t(\phi_i^t) = 0$. Hence, with the first-order surrogate adopted in IncrePA-ncvx and the incremental majorization-minimization scheme, we can apply Lemma 3 for the sequence $\hat{f}(x^t)$ and x^t to conclude that IncrePA-ncvx is almost sure convergent to the asymptotic stationary point of proximal approximation function $\hat{f}(x)$ with probability one. □

6 Experiments

In this section, we evaluate the empirical performance of IncrePA for both convex composite penalty and nonconvex composite penalty. We implemented the proposed method and all other methods for comparison in MATLAB. All experiments were conducted on a single core of a laptop and 2.6-GHz Intel CPU with 16 GB of RAM. We used both synthetic datasets and four real datasets[1] in the experiment. The real datasets are summarized in Table 1. We randomly sampled 80% of the data as training set and the rest as testing set. We used four different tasks to demonstrate the performance of the proposed method according to the convexity or nonconvexity of the composite penalty and general or strong convexity of the loss function. As a result, we provided empirical evidence for all kinds of combinations of loss functions and penalties, to which the proposed IncrePA has provided theoretical convergence results in the previous sections. In the following, we have:

- Section 6.1 considers a general convex loss with convex composite penalty by solving smooth hinge loss with the graph-guided lasso task on four real datasets;
- Section 6.2 considers a strongly convex loss with convex composite penalty by solving logistic loss with the large margin graph-guided lasso on four real datasets;
- Section 6.3 considers a nonconvex composite penalty of capped ℓ_1 norm overlapping group lasso on synthetic datasets with different numbers of groups and data points.
- Section 6.4 considers a nonconvex composite penalty of capped ℓ_1 norm graph-guided lasso on four real datasets.

6.1 Experiment 1: Solving general convex loss function with convex composite penalty

In this and the next subsections, we evaluate the performance of IncrePA on convex composite penalties in comparison with two incremental gradients ADMM: SA-ADMM (Zhong and Kwok 2014a) and SDCA-ADMM (Suzuki 2014) along with a PA-based stochastic gradient PA-ASGD (Zhong and Kwok 2014c). We do not consider the batch gradient PA method for

[1] 'a9a' and 'covtype' are from LIBSVM archive; 'protein' is from KDD CUP 2004; '20 newsgroup' is from http://www.cs.nyu.edu/~roweis/data.html.

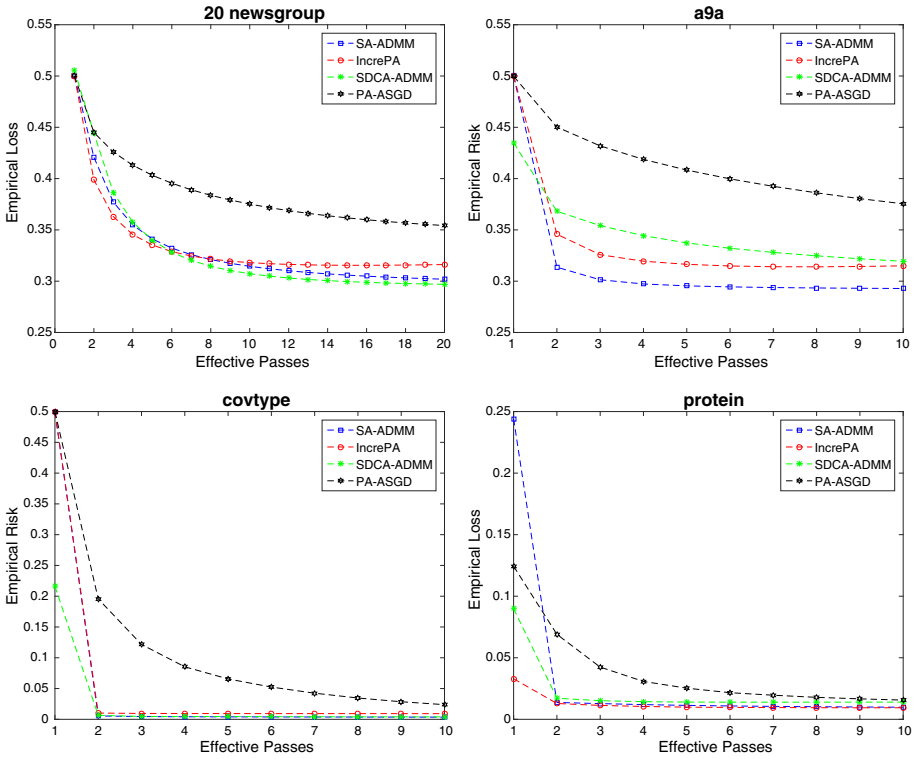


Fig. 1 General convex loss with convex composite penalty: empirical risk on training data versus effective passes of smooth hinge loss with graph-guided lasso on four real datasets, i.e. *upper left* 20 newsgroup, *upper right* a9a, *bottom left* covtype, and *bottom right* protein

comparison because Zhong and Kwok (2014c) has already shown that it is less efficient than PA-ASGD. Also, we do not explicitly compare the proposed algorithm with the stochastic ADMM methods because the latter is slower than the incremental ADMM methods as demonstrated in Zhong and Kwok (2014a) and Suzuki (2014).

In this subsection, we consider the general convex loss problem by using the smoothed hinge loss:

$$l_i(u) = \begin{cases} 0, & y_i u \geq 1 \\ \frac{1}{2} - y_i u, & y_i u \leq 0 \\ \frac{1}{2}(1 - y_i u)^2, & \text{otherwise}, \end{cases} \tag{34}$$

where $u = \xi_i^T x$, (ξ_i, y_i) is the i-th data sample. We utilize the graph-guided fused lasso

$$\lambda \left(||x||_1 + \sum_{\{i,j\} \in E} |x_i - x_j| \right) \tag{35}$$

as the convex composite regularizer. We construct the graph by sparse inverse covariance matrix as used in Suzuki (2014) and set λ at 0.001. The proximal map for $||x||_1$ is simply

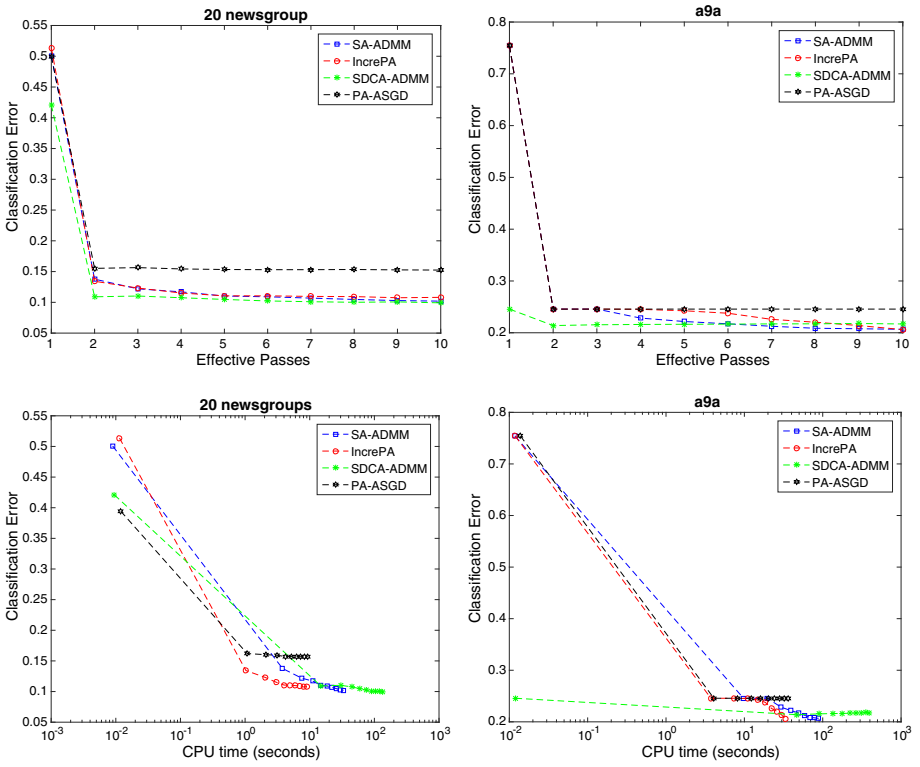


Fig. 2 General convex loss with convex composite penalty: classification error on testing data versus effective passes and CPU time of smooth hinge loss with graph-guided lasso on two real datasets, i.e. *upper left* 20 newsgroup (vs. effective passes), *upper right* a9a (vs. effective passes), *bottom left* 20 newsgroup (vs. CPU time), and *bottom right* a9a (vs. CPU time)

soft thresholding. The proximal map for $|x_i - x_j|$ is

$$[P^{\eta}_{r_k}]_s = \begin{cases} x_s - sign(x_i - x_j)\min\left\{\eta, \frac{|x_i - x_j|}{2}\right\}, & s \in \{i, j\} \\ x_s, & \text{otherwise} \end{cases} \tag{36}$$

as given in Yu (2013), Zhong and Kwok (2014c). For training performance, we report the empirical risk, which is the training loss, against the number of iterations for all datasets. As for the generalization performance, we report the classification error measured on testing set against the number of iterations and CPU time for the '20 newsgroup' and 'a9a', depicted in Fig. 2.

As shown in Fig. 1, in terms of reducing the empirical loss, the performance of the proposed method is the best on '20 newsgroup' and 'protein', and only inferior to SDCA-ADMM on 'covtype' and is only inferior to SA-ADMM on 'a9a'. On all datasets, IncrePA is more efficient than another PA-based method: PA-ASGD. Therefore, in this task, IncrePA performs almost the same as the other two ADMM-based incremental gradient methods and is a much faster PA-based method compared with PA-ASGD. Figure 2 demonstrates the generalization performance. When compared against the iteration numbers, IncrePA performs similar to SA-ADMM, which is better than PA-ASGD, although both are somewhat inferior to SDCA-ADMM. When compared against CPU time, the proposed method performs relatively better

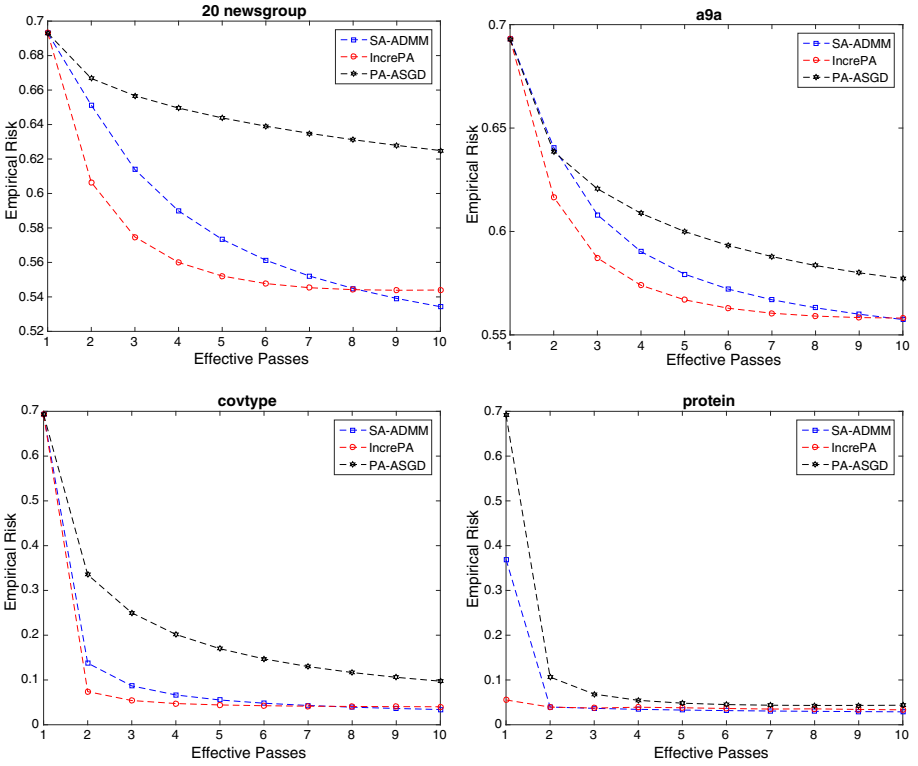


Fig. 3 Strongly convex loss with convex composite penalty: empirical risk on training data versus effective passes of logistic loss with large margin graph-guided lasso on four real datasets, i.e. *upper left* 20 newsgroup, *upper right* a9a, *bottom left* covtype, and *bottom right* protein

than the other methods on both datasets. As a conclusion, IncrePA has similar generalization performance in terms of classification error on both datasets with SA-ADMM and SDCA-ADMM and is more efficient than PA-ASGD. Also, the classification error on both testing sets indicate that our solution obtained by the surrogate regularizer is able to achieve satisfactory generalization performance.

6.2 Experiment 2: Solving strongly convex loss function with convex composite penalty

For the strongly convex case, we utilize logistic loss with the large margin graph-guided lasso regularizer as in Zhong and Kwok (2014c), i.e.

$$\lambda \left(||x||_2^2 + \sum_{\{i,j\} \in E} |x_i - x_j| \right). \tag{37}$$

We combine the logistic loss and the l_2 norm together to ensure the strong convexity of the loss part. Note that, in this case, the l_2 norm term can neither be incorporated into $l_i(\xi_i^T x)$, nor into $||Ax||_1$ form in the dual form, thus SDCA-ADMM is unable to handle this case because the dual problem does not fit ADMM structure. We only compare with the other two

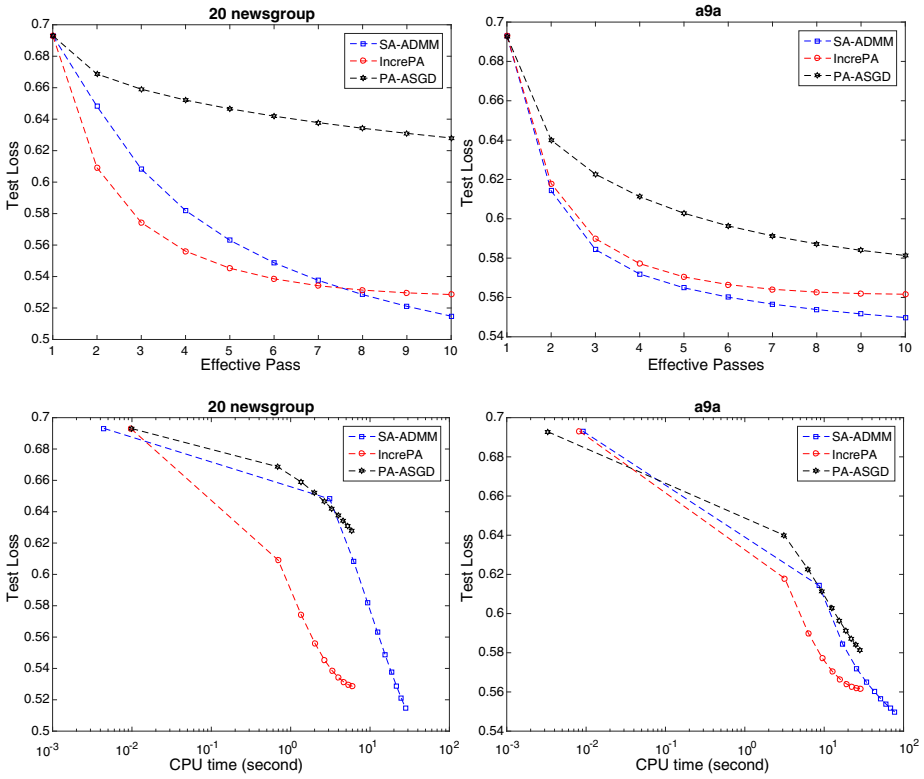


Fig. 4 General convex loss with convex composite penalty: test loss on testing data versus effective passes and CPU time of logistic loss with large margin graph-guided lasso on two real datasets, i.e. *upper left* 20 newsgroup (vs. effective passes), *upper right* a9a (vs. effective passes), *bottom left* 20 newsgroup (vs. CPU time), and *bottom right* a9a (vs. CPU time)

methods. We report the training loss on four real datasets, and testing loss versus iteration number and CPU time, respectively, for this case on '20 newsgroup' and 'a9a' datasets.

According to Fig. 3, our method performs relatively better on '20 newsgroup' and 'a9a', and is similar to SA-ADMM on 'covtype' and 'protein' in training. As for generalization performance, Fig. 4 shows the decrease of test loss over iteration number and CPU time. IncrePA performs better than the other two methods on '20 newsgroup' in terms of both number of iterations and CPU time. IncrePA is the best on 'a9a' in terms of CPU time, but falls behind SA-ADMM in terms of the number of iterations. Therefore, we conclude that IncrePA works comparably to ADMM-based incremental methods and is much better than the PA-based PA-ASGD method.

Before proceeding to the nonconvex composite penalty experiments, we would like to point out that, for the convex composite penalty, as a PA method, the proposed method has generally better performance than the stochastic gradient-based method: PA-ASGD, in terms of all performance metrics we have tried so far. As an incremental gradient-based method, the proposed method has comparable performance with SDCA-ADMM and SA-ADMM, but the merit of the proposed method is twofold: (1) The convergence analysis of SDCA-ADMM relies on the local strong convexity of the loss function. In addition, SDCA-ADMM requires that the dual problem should be in the structure for ADMM to be applied, which causes a

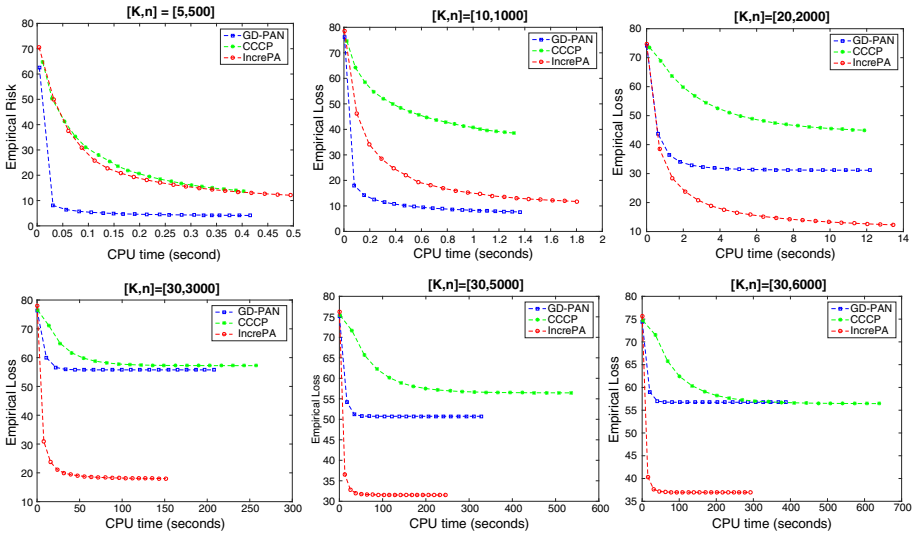


Fig. 5 Nonconvex composite penalty: empirical risk on training data versus CPU time of least square loss with capped ℓ_1 norm overlapping group lasso on synthetic datasets, i.e. *upper left* [K, n] = [5, 500], *upper middle* [K, n] = [10, 1000], *upper right* [K, n] = [20, 2000], *bottom left* [K, n] = [30, 3000], *bottom middle* [K, n] = [30, 5000], and *bottom right* [K, n] = [30, 6000]

stricter problem format and therefore limits its application domain. For example, in the above case, SDCA-ADMM cannot work at all because the dual parts do not fit into the structure for ADMM when being put together, despite each dual of their primal correspondence being easy to take. By contrast, the proposed method has given the convergence analysis for both general convex loss and strongly convex loss problems. Further, the format of the objective function in the proposed method is more general than SDCA-ADMM; (2) SA-ADMM lacks convergence analysis for the strongly convex loss problem, but the proposed one does.

6.3 Experiment 3: Solving nonconvex composite penalty of capped ℓ_1 overlapping croup lasso

This subsection studies the efficiency of IncrePA by comparing it with the other two algorithms, i.e. GD-PAN and CCCP, for such nonconvex composite penalty.

In this experiment, we consider capped-ℓ_1 norm coupled nonconvex overlapping group lasso:

$$\min_{x\in\mathbb{R}^d} \frac{1}{2n}||y - Sx||_2^2 + \lambda \sum_{k=1}^{K} \min\{||x_{g_k}||, \theta\}. \tag{38}$$

We use a synthetic data generated in the same style as Yu (2013). Specifically, the data s_i is generated independently and identically distributed from the normal distribution $\mathcal{N}(0, 1)$. The ground truth parameter x^* is generated as $x_j^* = (-1)^j exp(-\frac{j-1}{100})$. Therefore the dimension is $d = 90K + 10$ features. We set $y_i = (x^*)^T s_i + \vartheta_i$, where $\vartheta_i = 10\mathcal{N}(0, 1)$. We use the following pairs of (K, n) with both growing dimension and data number: (5, 500), (10, 1000), (20, 2000), (30, 3000), (30, 5000), (30, 6000). We also fix the dimension to $K = 30$ and increase the data number with $n = 4000, 5000, 6000, 8000$. We compare the proposed algorithm with GD_PAN and CCCP. For the GD_PAN method, we use the

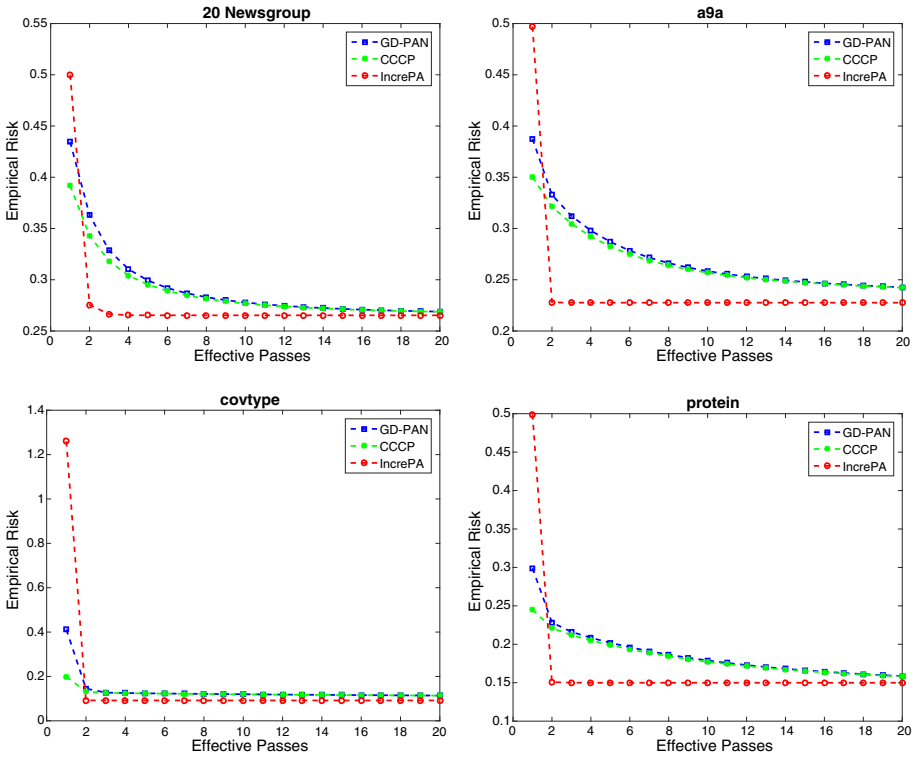


Fig. 6 Empirical risk on training data versus effective passes with nonconvex graph-guided lasso on four real datasets, i.e. *upper left* 20 newsgroup, *upper right* a9a, *bottom left* covtype, and *bottom right* protein

step size as suggested in Zhong and Kwok (2014b). We fix the parameters for all different methods to be $(\lambda, \theta) = (K/10, 0.1)$. For the proposed method, we choose the step size to obtain the largest descent in one pass over 5% of the data as suggested in Mairal (2014). We run each algorithm 10 times. The performance of the three algorithms is shown in Fig. 5 by plotting the objective value over CPU time. When the dataset is small, e.g. $(K, n) = (5, 500), (10, 1000)$, GD_PAN actually works better than the proposed method. However, when the dataset becomes large, the proposed method is much better, which indicates that the proposed method has better scalability.

6.4 Experiment 4: Solving nonconvex composite penalty of capped ℓ_1 graph-guided lasso

In this subsection, we considered nonconvex composite penalty by implementing the capped ℓ_1-norm with graph-guided lasso penalty on four real datasets (see Table 1). The graph is again constructed by sparse inverse covariance matrix. Again, we compared the proposed algorithm with GD-PAN and CCCP. We report the training efficiency in terms of training loss (objective value) over the effective pass of data in Fig. 6. It can be seen that the proposed method is consistently better than GD-PAN and CCCP in terms of training. For the '20 Newsgroup', 'a9a' and 'protein' datasets, the proposed method is much faster than the other two methods, while these three methods perform closely on the 'covtype'. We also reported

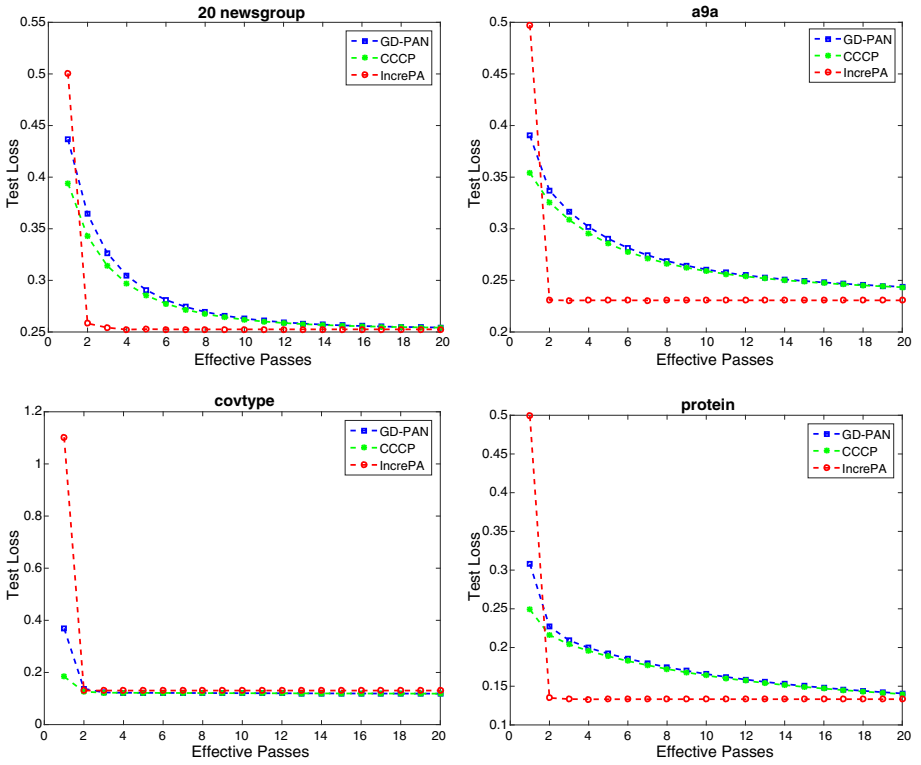


Fig. 7 Test loss on testing data versus effective passes with nonconvex graph-guided lasso on four real datasets, i.e. *upper left* 20 newsgroup, *upper right* a9a, *bottom left* covtype, and *bottom right* protein

test loss over the effective passes to show the generalization performance of the learned variable in Fig. 7. It can be seen that the proposed algorithm is more advantageous compared with both GD-PAN and CCCP on '20 newsgroup', 'a9a' and 'protein' datasets, while all of them perform similar on 'covtype' dataset. To sum up, the proposed method is more efficient than GD-PAN and CCCP in both training and testing.

7 Conclusion

In this paper, we have proposed a new incremental gradient method for empirical risk minimization regularized by composite regularizer. As a PA technique-based method, it is more efficient and faster than its existing batch and stochastic counterparts. When applied to convex composite penalties, compared with the popular ADMM-based incremental gradient, it has comparable performance, yet enjoys more compact update form and simpler theoretical analysis by virtue of the PA technique. Also, we have proposed a variant for nonconvex composite penalties, which has better scalability than the existing PA-based methods. Experimental results on four real datasets have shown its efficiency and satisfactory generalization performance for convex composite penalties. Further, experiments on both synthetic and real datasets has demonstrated its superior scalability and improved efficiency for nonconvex composite penalties.

Acknowledgements This work was supported by the Faculty Research Grant of Hong Kong Baptist University (HKBU) with the Project Codes: FRG2/14-15/075 and FRG2/15-16/049, by the National Natural Science Foundation of China with the Grant Numbers: 61272366 and 61672444, and by the SZSTI Grant: JCYJ20160531194006833.

Appendix: Proof of Theorems 1 and 2

The proof of Theorems 1 and 2 is a combination of the proof in Defazio et al. (2014a) and Yu (2013). We proceed the proof by two steps. First we prove that for the PA approximation $\hat{f}(x) = l(x) + \hat{r}(x)$ and its global optimal value $\hat{f}(\hat{x}^*)$, $\hat{f}(x^t) - \hat{f}(\hat{x}^*)$ converges linearly in expectation. Then we conclude the proof using Lemma 1 which states that the surrogate $\hat{f}(x)$ and the original $F(x)$ can be arbitrarily close.

For the first step, we use the following Lyapunov function,

$$T^t = Q^t + \left(c_1 + \frac{c_2}{\eta}\right)||x^t - \hat{x}^*||_2^2 + c_2\big(\hat{f}(x^t) - \hat{f}(\hat{x}^*)\big), \tag{39}$$

$$Q^t = \frac{1}{n}\sum_{i=1}^{n} l_i(\phi_i^t) - l(\hat{x}^*) - \frac{1}{n}\sum_{i=1}^{n}\left\langle \nabla l_i(\hat{x}^*), \phi_i^t - \hat{x}^*\right\rangle, \tag{40}$$

where η is the step size and c_1, c_2 are constants to be specified in the following proof. Since the step size η is related to ϵ, our choice of parameters should be different from Defazio et al. (2014a), otherwise, the difference between $\hat{f}(x)$ and $F(x)$ cannot be small enough. We borrow techniques and intermediate results from Defazio et al. (2014a), which could be summarized into the following three lemmas. We omit the proof of these lemmas as they paraphrase those found in Defazio et al. (2014a). In the following, the expectation is conditioned on information up to iteration t, where i^t is the randomly picked index, thus is a random variable.

Lemma 4 (Defazio et al. 2014a, Theorem 1) $\mathbb{E}[Q^{t+1}]$ *has the following iterative relationship with* Q^t,

$$\mathbb{E}[Q^{t+1}] = \left(1 - \frac{1}{n}\right)Q^t + \frac{1}{n}\left[l(x^t) - l(\hat{x}^*) - \left\langle \nabla l(\hat{x}^*), x^t - \hat{x}^*\right\rangle\right]. \tag{41}$$

The next lemma bounds $\mathbb{E}c_1||x^{t+1} - \hat{x}^*||_2^2$. Recall that $\hat{x}^*$ is the optimal point of the approximate function $\hat{f}(x)$, we have:

Lemma 5 (Defazio et al. 2014a, Theorem 1)

$$\begin{aligned} c_1\mathbb{E}||x^{t+1} - \hat{x}^*||_2^2 \le{}& (1 - \eta\mu)c_1||x^t - \hat{x}^*||_2^2 + 2(1 + \beta^{-1})c_1\eta^2 L Q^t \\ &+ \big((1 + \beta)c_1\eta^2 - \frac{c_1\eta}{L}\big)\mathbb{E}\big[||\nabla l_{i^t}(x^t) - \nabla l_{i^t}(\hat{x}^*)||_2^2\big] \\ &- \left(2c_1\eta^2\beta\mu + 2c_1\eta\left(1 - \frac{\mu}{L}\right)\right)\left[l(x^t) - l(\hat{x}^*) - \left\langle \nabla l(\hat{x}^*), x^t - \hat{x}^*\right\rangle\right]. \end{aligned} \tag{42}$$

Finally we prepare ourselves with the lemma bounding $\mathbb{E}\left[c_2\big(\hat{f}(x^{t+1}) - \hat{f}(\hat{x}^*)\big) + \frac{c_2}{\eta}||x^{t+1} - \hat{x}^*||_2^2\right]$.

Lemma 6 (Defazio et al. 2014a, Theorem 2) *For some* $\beta > 0$,

$$\begin{aligned} c_2\mathbb{E}\big[\hat{f}(x^{t+1}) - \hat{f}(\hat{x}^*)\big] + \frac{c_2}{2\eta}\mathbb{E}||x^{t+1} - \hat{x}^*||_2^2 \le{}& \frac{c_2}{2\eta}||x^t - \hat{x}^*||_2^2 \\ &+ 2(1 + \beta^{-1})c_2\eta L Q^t + (1 + \beta)c_2\eta\mathbb{E}||\nabla l_{i^t}(x^t) - \nabla l_{i^t}(\hat{x}^*)||_2^2. \end{aligned} \tag{43}$$

Proof of Theorem 1: General convex loss function case

With Lemmas 1–3, we have:

$$\begin{aligned}\mathbb{E}[T^{t+1}] &= \mathbb{E}\left[Q^{t+1} + \left(c_1 + \frac{c_2}{2\eta}\right)||x^{t+1} - \hat{x}^*||_2^2 + c_2[\hat{f}(x^{t+1}) - \hat{f}(\hat{x}^*)]\right] \\ &\leq \left[Q^t + \left(c_1 + \frac{c_2}{2\eta}\right)||x^t - \hat{x}^*||_2^2\right] \\ &\quad + \left(2(1+\beta^{-1})\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 L - \frac{1}{n}\right)Q^t \\ &\quad + \left((1+\beta)\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 - \frac{c_1\eta}{L}\right)\mathbb{E}\left[||\nabla l_{i^t}(x^t) - \nabla l_{i^t}(\hat{x}^*)||_2^2\right] \\ &\quad + \left(\frac{1}{n} - 2c_1\eta\right)\left[l(x^t) - l(\hat{x}^*) - \left\langle\nabla l(\hat{x}^*), x^t - \hat{x}^*\right\rangle\right].\end{aligned} \tag{44}$$

With the choice of parameters $c_1, c_2, \frac{1}{\kappa}, \beta$ and η, the terms in big round brackets are non-positive (we defer this discussion to the end of the proof). We leave out these non-positive terms,

$$\mathbb{E}\left[Q^{t+1} + \left(c_1 + \frac{c_2}{2\eta}\right)||x^{t+1} - \hat{x}^*||_2^2 + c_2[\hat{f}(x^{t+1}) - \hat{f}(\hat{x}^*)]\right] \leq \left[Q^t + \left(c_1 + \frac{c_2}{2\eta}\right)||x^t - \hat{x}^*||_2^2\right]. \tag{45}$$

By this iteration relationship and considering from step 0 to t, we have

$$\begin{aligned}\mathbb{E}c_2\left[\hat{f}(\bar{x}^t) - \hat{f}(\hat{x}^*)\right] &\leq \frac{1}{t}\mathbb{E}\left[c_2\sum_{i=0}^{t}[\hat{f}(x^i) - \hat{f}(\hat{x}^*)]\right] \\ &\leq \frac{1}{t}\left(Q^0 + \left(c_1 + \frac{c_2}{2\eta}\right)||x^0 - \hat{x}^*||_2^2\right. \\ &\quad \left. - \mathbb{E}\left[Q^t + \left(c_1 + \frac{c_2}{2\eta}\right)||x^t - \hat{x}^*||_2^2\right]\right) \\ &\leq \frac{1}{t}\left(Q^0 + \left(c_1 + \frac{c_2}{2\eta}\right)||x^0 - \hat{x}^*||_2^2\right),\end{aligned} \tag{46}$$

where

$$\begin{aligned}Q^0 &= \frac{1}{n}\sum_{i=1}^{n} l_i(\phi_i^0) - l(\hat{x}^*) - \frac{1}{n}\sum_{i=1}^{n}\left\langle\nabla l_i(\hat{x}^*), \phi_i^0 - \hat{x}^*\right\rangle \\ \bar{x}^t &= \frac{1}{t}\sum_{i=1}^{t} x^i.\end{aligned} \tag{47}$$

Hence as long as $t \geq \frac{1}{c_2\epsilon}\left(Q^0 + \left(c_1 + \frac{c_2}{2\eta}\right)||x^0 - \hat{x}^*||_2^2\right)$, we get:

$$\mathbb{E}\left[\hat{f}(\bar{x}^t) - \hat{f}(\hat{x}^*)\right] \leq \epsilon. \tag{48}$$

We have (i) $\mathbb{E}[F(\bar{x}^t) - \hat{f}(\bar{x}^t)] \leq \epsilon$ by Lemma 1 (i.e. $r(x) - \hat{r}(x) \leq \epsilon,\ \forall x$); (ii) $\mathbb{E}[\hat{f}(\bar{x}^t) - \hat{f}(\hat{x}^*)] \leq \epsilon$ by inequality (48); (iii) $\mathbb{E}[\hat{f}(\hat{x}^*) - F(\hat{x}^*)] \leq 0$ by Lemma 1 (i.e. $0 \leq r(x) - \hat{r}(x),\ \forall x$). By (i)–(iii), we have

$$\begin{aligned}\mathbb{E}\big[F(\bar{x}^t) - F(\hat{x}^*)\big] &= \mathbb{E}\big[\big(F(\bar{x}^t) - \hat{f}(\bar{x}^t)\big) + \big(\hat{f}(\bar{x}^t) - \hat{f}(\hat{x}^*)\big) + \big(\hat{f}(\hat{x}^*) - F(\hat{x}^*)\big)\big] \\ &\le \epsilon + \epsilon + 0.\end{aligned} \tag{49}$$

Thus, as long as $t \ge \frac{1}{c_2\epsilon}\left(Q^0 + \left(c_1 + \frac{c_2}{2\eta}\right)||x^0 - \hat{x}^*||_2^2\right)$, $\mathbb{E}\big[F(\bar{x}^t) - F(\hat{x}^*)\big] \le 2\epsilon$.

Verification of non-positiveness for general convex case in (44)

In the following, we show the non-positiveness of the terms in the round bracket in (44). For the four inequalities, when $\eta < \min(\frac{1}{2L}, \frac{2\epsilon}{M^2})$, $c_1 = \frac{1}{2\eta n}$, $c_2 = \frac{1}{2n}\left(\frac{1}{2\eta L\beta} - 1\right)$, $\beta = 1$, we need to ensure,

$$I_1\text{: } 2(1+\beta^{-1})\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 L - \frac{1}{n} \le 0, \tag{50}$$

$$I_2\text{: } (1+\beta)\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 - \frac{c_1\eta}{L} \le 0, \tag{51}$$

$$I_3\text{: } \frac{1}{n} - 2c_1\eta \le 0. \tag{52}$$

I_3 is 0 when $c_1 = \frac{1}{2\eta n}$; $c_2 = \frac{1}{2n}\left(\frac{1}{2\eta L\beta} - 1\right) = c_1\eta\left(\frac{1}{2\eta L\beta} - 1\right) \le c_1\eta\left(\frac{1}{\eta L(1+\beta)} - 1\right)$, thus I_2 is satisfied. Substituting c_1 and c_2 into I_1, gives $(1+\frac{1}{\beta})\frac{1}{\beta} \le 1$, meanwhile, $2\eta L\frac{1}{\beta} \le 1$, both of which are satisfied when $\beta = 1$ under $\eta < \frac{1}{2L}$.

Proof of Theorem 2: Strongly convex loss function case

Combining Lemmas 1–3 and rearranging the term, we can get:

$$\begin{aligned}\mathbb{E}[T^{t+1}] &= \mathbb{E}\left[Q^{t+1} + \left(c_1 + \frac{c_2}{2\eta}\right)||x^{t+1} - \hat{x}^*||_2^2 + c_2[\hat{f}(x^{t+1}) - \hat{f}(\hat{x}^*)]\right] \\ &\le \left[Q^t + \left(c_1 + \frac{c_2}{2\eta}\right)||x^t - \hat{x}^*||_2^2\right] \\ &\quad + \left(2(1+\beta^{-1})\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 L - \frac{1}{n}\right)Q^t - \eta\mu c_1||x^t - \hat{x}^*||_2^2 \\ &\quad + \left((1+\beta)\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 - \frac{c_1\eta}{L}\right)\mathbb{E}\big[||\nabla l_{i^t}(x^t) - \nabla l_{i^t}(\hat{x}^*)||_2^2\big] \\ &\quad + \left(\frac{1}{n} - 2c_1\eta^2\beta\mu - 2c_1\eta\left(1 - \frac{\mu}{L}\right)\right)\big[l(x^t) - l(\hat{x}^*) - \langle\nabla l(\hat{x}^*), x^t - \hat{x}^*\rangle\big].\end{aligned} \tag{53}$$

Adding a nonnegative term $c_2\left(1 - \frac{1}{\kappa}\right)[\hat{f}(x^t) - \hat{f}(\hat{x}^*)]$ to the RHS with $\frac{1}{\kappa} \in (0, 1)$, we have:

$$\begin{aligned}\mathbb{E}[T^{t+1}] &\le T^t - \frac{c_2}{\kappa}[\hat{f}(x^t) - \hat{f}(\hat{x}^*)] \\ &\quad + \left(2(1+\beta^{-1})\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 L - \frac{1}{n}\right)Q^t - \eta\mu c_1||x^t - \hat{x}^*||_2^2 \\ &\quad + \left((1+\beta)\left(c_1 + \frac{c_2}{\eta}\right)\eta^2 - \frac{c_1\eta}{L}\right)\mathbb{E}\big[||\nabla l_{i^t}(x^t) - \nabla l_{i^t}(\hat{x}^*)||_2^2\big] \\ &\quad + \left(\frac{1}{n} - 2c_1\eta^2\beta\mu - 2c_1\eta\left(1 - \frac{\mu}{L}\right)\right)\big[l(x^t) - l(\hat{x}^*) - \langle\nabla l(\hat{x}^*), x^t - \hat{x}^*\rangle\big].\end{aligned} \tag{54}$$

After further extracting a $-\frac{1}{\kappa}T^t$ term from the RHS, we have

$$\begin{aligned}\mathbb{E}\left[T^{t+1}\right]-T^t \leq & -\frac{1}{\kappa}T^t+\left(\frac{1}{\kappa}+2(1+\beta^{-1})\left(c_1+\frac{c_2}{\eta}\right)\eta^2 L-\frac{1}{n}\right)Q^t \\ & +\left(\frac{1}{\kappa}\left(c_1+\frac{c_2}{2\eta}\right)-\eta\mu c_1\right)||x^t-\hat{x}^*||_2^2 \\ & +\left((1+\beta)\left(c_1+\frac{c_2}{\eta}\right)\eta^2-\frac{c_1\eta}{L}\right)\mathbb{E}\left[||\nabla l_{i^t}(x^t)-\nabla l_{i^t}(\hat{x}^*)||_2^2\right] \\ & +\left(\frac{1}{n}-2c_1\eta^2\beta\mu-2c_1\eta\left(1-\frac{\mu}{L}\right)\right)\left[l(x^t)-l(\hat{x}^*)-\left\langle\nabla l(\hat{x}^*),x^t-\hat{x}^*\right\rangle\right].\end{aligned} \tag{55}$$

With the choice of parameters $c_1, c_2, \frac{1}{\kappa}, \beta$ and η, the terms in big round brackets are non-positive (we defer this discussion to the end of the proof). We leave out these non-positive terms and by the iteration relation,

$$\begin{aligned}\mathbb{E}\left[\hat{f}(x^t)-\hat{f}(\hat{x}^*)\right] &\leq \mathbb{E}\left[T^t\right]\leq\left(1-\frac{1}{\kappa}\right)^t T^0 \\ &=\left(1-\frac{1}{\kappa}\right)^t\left[Q^0+\left(c_1+\frac{c_2}{\eta}\right)||x^0-\hat{x}^*||_2^2+c_2\left(\hat{f}(x^0)-\hat{f}(\hat{x}^*)\right)\right],\end{aligned} \tag{56}$$

where

$$Q^0=\frac{1}{n}\sum_{i=1}^n l_i(\phi_i^0)-l(\hat{x}^*)-\frac{1}{n}\sum_{i=1}^n\left\langle\nabla l_i(\hat{x}^*),\phi_i^0-\hat{x}^*\right\rangle. \tag{57}$$

Hence, as long as $t\geq\log\frac{T^0}{\epsilon}\big/\log(1-\frac{1}{\kappa})$, we have:

$$\mathbb{E}\left[\hat{f}(x^t)-\hat{f}(\hat{x}^*)\right]\leq\epsilon. \tag{58}$$

Similar to the reasoning in the proof of Theorem 1 and by Lemma 1,

$$\begin{aligned}\mathbb{E}\left[F(x^t)-F(\hat{x}^*)\right]&=\mathbb{E}\left[\left(F(x^t)-\hat{f}(x^t)\right)+\left(\hat{f}(x^t)-\hat{f}(\hat{x}^*)\right)+\left(\hat{f}(\hat{x}^*)-F(\hat{x}^*)\right)\right] \\ &\leq\epsilon+\epsilon+0.\end{aligned} \tag{59}$$

Finally, we conclude that, as long as $t\geq\log\frac{T^0}{\epsilon}\big/\log(1-\frac{1}{\kappa})$, we have

$$\mathbb{E}\left[F(x^t)-F(\hat{x}^*)\right]\leq 2\epsilon.$$

Verification of non-positiveness for strongly convex case in (55)

In the following, we show the non-positiveness of the terms in the round bracket in (55). For the four inequalities, when $\eta<\min(\frac{1}{2L},\frac{2\epsilon}{M^2})$, $c_1=\frac{1}{2\eta n}\frac{L}{L-\mu}$, $c_2=c_1\eta\left(\frac{1}{2\eta L\beta}-1\right)$, $\frac{1}{\kappa}=\frac{2\eta\mu}{1+\frac{1}{2\eta L}}$, we shall verify

$$I_1\text{: }\frac{1}{\kappa}+2(1+\beta^{-1})\left(c_1+\frac{c_2}{\eta}\right)\eta^2 L-\frac{1}{n}\leq 0, \tag{60}$$

$$I_2\text{: }\frac{1}{\kappa}\left(c_1+\frac{c_2}{2\eta}\right)-\eta\mu c_1\leq 0, \tag{61}$$

$$I_3\text{: }(1+\beta)\left(c_1+\frac{c_2}{\eta}\right)\eta^2-\frac{c_1\eta}{L}\leq 0, \tag{62}$$

$$I_4\colon \frac{1}{n} - 2c_1\eta^2\beta\mu - 2c_1\eta\left(1-\frac{\mu}{L}\right) \le 0. \tag{63}$$

First consider I_4, when $c_1 = \frac{1}{2\eta n}\frac{L}{L-\mu}$,

$$c_1 = \frac{1}{2\eta n}\frac{L}{L-\mu} > \frac{1}{2\eta n}\frac{1}{1-\frac{\mu}{L}+\mu\eta\beta}. \tag{64}$$

Thus, I_4 is satisfied:

$$\frac{1}{n} - 2c_1\eta^2\beta\mu - 2c_1\eta\left(1-\frac{\mu}{L}\right) \le 0. \tag{65}$$

Next for I_3, under $c_2 = \frac{1}{2n}\frac{L}{L-\mu}\left(\frac{1}{2\eta L\beta}-1\right)$ and the fact $\beta > 1$,

$$c_2 = c_1\eta\left(\frac{1}{2\eta L\beta}-1\right) \le c_1\eta\left(\frac{1}{(1+\beta)\eta L}-1\right), \tag{66}$$

which is equal to I_3, thus we have verified I_3. In addition, we need $2\eta L\beta < 1$ to guarantee $c_2 > 0$ which is satisfied when substituting β in.

Now we move to I_2, as $\frac{1}{\beta} < 1$ and $\frac{1}{\kappa} = \frac{2\eta\mu}{1+\frac{1}{2\eta L}}$,

$$\frac{1}{\kappa} = \frac{2\eta\mu}{1+\frac{1}{2\eta L}\cdot 1} \le \frac{2\eta\mu}{1+\frac{1}{2\eta L}\cdot\frac{1}{\beta}}, \tag{67}$$

which is equivalent to I_2 after we substitute c_1, c_2 into I_2. Apparently, as $2\eta L\beta < 1$, $2\eta\mu \le 2\eta L \le 2\eta L\beta < 1$ and $1+\frac{1}{2\eta L} \ge 1+\frac{1}{2\eta L\beta} > 2$, thus $\frac{1}{\kappa} < 1$ is also satisfied.

Finally, we deal with I_1. When $\eta < \frac{1}{2L}$, we have $\eta \le \frac{1}{2L} + \frac{1}{4L^2}\frac{2n\mu 2(L-\mu)}{L}$, given $\eta < \frac{1}{2n\mu}$, which is equivalent to

$$4L^2\eta^2 + \left(2L + \frac{2n\mu 2(L-\mu)}{L}\right)\eta \le 0 \le 2\left(1-\frac{\mu}{L}\right). \tag{68}$$

Rearranging the terms, we get

$$(1+2\eta L)2\eta L \le (1-2n\eta\mu)\,2\left(1-\frac{\mu}{L}\right). \tag{69}$$

Note the fact that $(1-2n\eta\mu)\,2\left(1-\frac{\mu}{L}\right) < \left(1-\frac{2n\eta\mu 2\eta L}{2\eta L+1}\right)2\left(1-\frac{\mu}{L}\right) < 2$. We have $(1+2\eta L)2\eta L$ is strictly less than $\left(1-\frac{2\eta\mu 2\eta L}{2\eta L+1}\right)2\left(1-\frac{\mu}{L}\right)$ under $\eta < \frac{1}{2L}$. After we substituting $\frac{1}{\kappa}, c_1, c_2$ into I_1,

$$\left(1+\frac{1}{\beta}\right)\frac{1}{\beta} \le \left(1-\frac{2\eta\mu n 2\eta L}{2\eta L+1}\right)2\left(1-\frac{\mu}{L}\right), \tag{70}$$

with the additional requirement for β that $2\eta L < \frac{1}{\beta} \le 1$, we can guarantee that there exists some β (e.g. taking the average) to satisfy the relationship,

$$(1+2\eta L)2\eta L < \left(1+\frac{1}{\beta}\right)\frac{1}{\beta} \le \left(1-\frac{2\eta\mu n 2\eta L}{2\eta L+1}\right)2\left(1-\frac{\mu}{L}\right). \tag{71}$$

References

Azadi, S., & Sra, S. (2014). Towards an optimal stochastic alternating direction method of multipliers. In *Proceedings of the 31st international conference on machine learning* (pp. 620–628).

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

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

Borwein, J. M., & Lewis, A. S. (2010). *Convex analysis and nonlinear optimization: Theory and examples*. Berlin: Springer.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177–186). Berlin: Springer.

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

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014a). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*. arXiv:1407.0202.

Defazio, A., Domke, J., & Caetano, T. (2014b). Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the 31st international conference on machine learning (ICML-14)* (pp. 1125–1133).

Ghadimi, S., & Lan, G. (2012). Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. I: A generic algorithmic framework. *SIAM Journal on Optimization*, *22*(4), 1469–1492.

Gong, P., Zhang, C., Lu, Z., Huang, J., & Ye, J. (2013). A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proceedings of the 30th international conference on machine learning* (pp. 37–45).

Jacob, L., Vert, J., & Obozinski, G. R. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th international conference on machine learning (ICML-09)* (p. 55).

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

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

Konečnỳ, J., & Richtárik, P. (2013). Semi-stochastic gradient descent methods. arXiv:1312.1666.

Lacoste-Julien, S., Schmidt, M., & Bach, F. (2012). A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. arXiv:1212.2002.

Lu, Z. (2012). Sequential convex programming methods for a class of structured nonlinear programming. arXiv:1210.3039.

Mairal, J. (2014). Incremental majorization-minimization optimization with application to large-scale machine learning. arXiv:1402.4419.

Nesterov, Y., & Nesterov, I. U. E. (2004). *Introductory lectures on convex optimization: A basic course* (Vol. 87). London: Springer.

Ouyang, H., He, N., Tran, L., & Gray, A. (2013). Stochastic alternating direction method of multipliers. In *Proceedings of the 30th international conference on machine learning* (pp. 80–88).

Roux, N. L., Schmidt, M., & Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25, pp. 2663–2671). Newry: Curran Associates.

Shalev-Shwartz, S., & Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, *14*(1), 567–599.

Shamir, O., & Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th international conference on machine learning* (pp. 71–79).

Shen, X., & Huang, H.-C. (2010). Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, *105*(490), 727–739.

Suzuki, T. (2013). Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 392–400).

Suzuki, T. (2014). Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proceedings of the 31st international conference on machine learning* (pp. 736–744).

Xiang, S., Tong, X., & Ye, J. (2013). Efficient sparse group feature selection via nonconvex optimization. In *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 284–292).

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

Xiao, L., & Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. arXiv:1403.4699.

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

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*, 894–942.

Zhang, T. (2010). Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, *11*, 1081–1107.

Zheng, S., & Kwok J. T. (2016). Fast-and-light stochastic ADMM. arXiv:1604.07070.

Zhong, W., & Kwok, J. (2014a). Fast stochastic alternating direction method of multipliers. In *Proceedings of the 31st international conference on machine learning* (pp. 46–54).

Zhong, W., & Kwok, J. (2014b). Gradient descent with proximal average for nonconvex and composite regularization. In *AAAI conference on artificial intelligence*.

Zhong, L. W., & Kwok, J. T. (2014c). Accelerated stochastic gradient method for composite regularization. In *Proceedings of the seventeenth international conference on artificial intelligence and statistics* (pp. 1086–1094).