A Dual-Augmented Block Minimization Framework for Learning with Limited Memory

Ian E.H. Yen [*] **Shan-Wei Lin** [†] **Shou-De Lin** [†]
[*] University of Texas at Austin [†] National Taiwan University
[*] `ianyen@cs.utexas.edu` `{r03922067,sdlin}@csie.ntu.edu.tw`

Abstract

In past few years, several techniques have been proposed for training of linear Support Vector Machine (SVM) in limited-memory setting, where a dual block-coordinate descent (dual-BCD) method was used to balance cost spent on I/O and computation. In this paper, we consider the more general setting of regularized *Empirical Risk Minimization (ERM)* when data cannot fit into memory. In particular, we generalize the existing block minimization framework based on strong duality and *Augmented Lagrangian* technique to achieve global convergence for general convex ERM. The block minimization framework is flexible in the sense that, given a solver working under sufficient memory, one can integrate it with the framework to obtain a solver globally convergent under limited-memory condition. We conduct experiments on L1-regularized classification and regression problems to corroborate our convergence theory and compare the proposed framework to algorithms adopted from online and distributed settings, which shows superiority of the proposed approach on data of size ten times larger than the memory capacity.

1 Introduction

Nowadays data of huge scale are prevalent in many applications of statistical learning and data mining. It has been argued that model performance can be boosted by increasing both number of samples and features, and through crowdsourcing technology, annotated samples of terabytes storage size can be generated [3]. As a result, the performance of model is no longer limited by the sample size but the amount of available computational resources. In other words, the data size can easily go beyond the size of physical memory of available machines. Under this setting, most of learning algorithms become slow due to expensive I/O from secondary storage device [26].

When it comes to huge-scale data, two settings are often considered — online and distributed learning. In the online setting, each sample is processed only once without storage, while in the distributed setting, one has several machines that can jointly fit the data into memory. However, the real cases are often not as extreme as these two — there are usually machines that can fit part of the data, but not all of them. In this setting, an algorithm can only process a block of data at a time. Therefore, balancing the time spent on I/O and computation becomes the key issue [26]. Although one can employ an online-fashioned learning algorithm in this setting, it has been observed that online method requires large number of epoches to achieve comparable performance to batch method, and at each epoch it spends most of time on I/O instead of computation [2, 21, 26]. The situation for online method could become worse for problem of non-smooth, non-strongly convex objective function, where a qualitatively slower convergence of online method is exhibited [15, 16] than that proved for strongly-convex problem like SVM [14].

In the past few years, several algorithms have been proposed to solve large-scale linear Support Vector Machine (SVM) in the limited memory setting [2, 21, 26]. These approaches are based on a dual

Block Coordinate Descent (dual-BCD) algorithim, which decomposes the original problem into a series of block sub-problems, each of them requires only a block of data loaded into memory. The approach was proved linearly convergent to the global optimum, and demonstrated fast convergence empirically. However, the convergence of the algorithm relies on the assumption of a smooth dual problem, which, as we show, does not hold generally for other regularized *Empirical Risk Minimizaton (ERM)* problem. As a result, although the dual-BCD approach can be extended to the more general setting, it is not globally convergent except for a class of problems with L2-regularizer.

In this paper, we first show how to adapt the dual block-coordinate descnet method of [2, 26] to the general setting of regularized *Empirical Risk Mimization (ERM)*, which subsumes most of supervised learning problems ranging from classification, regression to ranking and recommendation. Then we discuss the convergence issue arises when the underlying ERM is not strongly-convex. A *Primal Proximal Point* (or *Dual Augmented Lagrangian*) method is then proposed to address this issue, which as we show, results in a block minimization algorithm with global convergence to optimum for convex regularized ERM problems. The framework is flexible in the sense that, given a solver working under sufficient-memory condition, it can be integrated into the block minimization framework to obtain a solver globally convergent under limited-memory condition.

We conduct experiments on L1-regularized classification and regression problems to corroborate our convergence theory, which shows that the proposed simple dual-augmented technique changes the convergence behavior dramatically. We also compare the proposed framework to algorithms adopted from online and distributed settings. In particular, we describe how to adapt a distributed optimization framework — Alternating Direction Method of Multiplier (ADMM) [1] — to the limited-memory setting, and show that, although the adapted algorithm is effective, it is not as efficient as the proposed framework specially designed for limited-memory setting. Note our experiment does not adapt into comparison some recently proposed distributed learning algorithms (CoCoA etc.) [7, 10] that only apply to ERM with L2-regularizer or some other distributed method designed for some specific loss function [19].

2 Problem Setup

In this work, we consider the regularized *Empirical Risk Minimization* problem, which given a data set $\mathcal{D} = \{(\Phi_n, \boldsymbol{y}_n)\}_{n=1}^N$, estimates a model through

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d, \boldsymbol{\xi}_n\in\mathbb{R}^p} \quad & F(\boldsymbol{w},\boldsymbol{\xi}) = R(\boldsymbol{w}) + \sum_{n=1}^{N} L_n(\boldsymbol{\xi}_n) \\ s.t. \quad & \Phi_n \boldsymbol{w} = \boldsymbol{\xi}_n,\ n \in [N] \end{aligned} \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the model parameter to be estimated, Φ_n is a p by d design matrix that encodes features of the n-th data sample, $L_n(\boldsymbol{\xi}_n)$ is a convex loss function that penalizes the discrepancy between ground truth and prediction vector $\boldsymbol{\xi}_n \in \mathbb{R}^p$, and $R(\boldsymbol{w})$ is a convex regularization term penalizing model complexity.

The formulation (1) subsumes a large class of statistical learning problems ranging from classification [27], regression [17], ranking [8], and convex clustering [24]. For example, in classification problem, we have $p = |\mathcal{Y}|$ where $\mathcal{Y}$ consists of the set of all possible labels and $L_n(\boldsymbol{\xi})$ can be defined as the logistic loss $L_n(\boldsymbol{\xi}) = \log(\sum_{k\in\mathcal{Y}} \exp(\xi_k)) - \xi_{y_n}$ as in logistic regression or the hinge loss $L_n(\boldsymbol{\xi}) = \max_{k\in\mathcal{Y}}(1 - \delta_{k,y_n} + \xi_k - \xi_{y_n})$ as used in support vector machine; in a (multi-task) regression problem, the target variable consists of K real values $\mathcal{Y} = \mathbb{R}^K$, the prediction vector has $p = K$ dimensions, and a square loss $L_n(\boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{y}_n\|_2^2$ is often used. There are also a variety of regularizers $R(\boldsymbol{w})$ employed in different applications, which includes the L2-regularizer $R(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2$ in ridge regression, L1-regularizer $R(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_1$ in *Lasso*, nuclear-norm $R(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_*$ in matrix completion, and a family of structured group norms $R(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_{\mathcal{G}}$ [11]. Although the specific form of $L_n(\boldsymbol{\xi})$, $R(\boldsymbol{w})$ does not affect the implementation of the limited-memory training procedure, two properties of the functions — strong convexity and smoothness — have key effects on the behavior of the block minimization algorithm.

Definition 1 (Strong Convexity)**.** *A function $f(x)$ is strongly convex iff it is lower bounded by a simple quadratic function*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{m}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \tag{2}$$

for some constant $m > 0$ and $\forall \boldsymbol{x}, \boldsymbol{y} \in dom(f)$.

Definition 2 (Smoothness)**.** *A function $f(x)$ is smooth iff it is upper bounded by a simple quadratic function*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{M}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \tag{3}$$

for some constant $0 \leq M < \infty$ and $\forall \boldsymbol{x}, \boldsymbol{y} \in dom(f)$.

For instance, the square loss and logistic loss are both smooth and strongly convex [1], while the hinge-loss satisfies neither of them. On the other hand, most of regularizers such as L1-norm, structured group norm, and nuclear norm are neither smooth nor strongly convex, except for the L2-regularizer, which satifies both. In the following we will demonstrate the effects of these properties to Block Minimization algorithms.

Throughout this paper, we will assume that a solver for (1) that works in sufficient-memory condition is given, and our task is to design an algorithmic framework that integrates with the solver to efficiently solve (1) when data cannot fit into memory. We will assume, however, that the d-dimensional parameter vector $\boldsymbol{w}$ can be fit into memory.

3 Dual Block Minimization

In this section, we extend the block minimization framework of [26] from linear SVM to the general setting of regularized ERM (1).The dual of (1) can be expressed as

$$\begin{aligned} \min_{\boldsymbol{\mu}\in\mathbb{R}^d, \boldsymbol{\alpha}_n\in\mathbb{R}^p} \quad & R^*(-\boldsymbol{\mu}) + \sum_{n=1}^{N} L_n^*(\boldsymbol{\alpha}_n) \\ s.t. \quad & \sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n = \boldsymbol{\mu} \end{aligned} \tag{4}$$

where $R^*(-\boldsymbol{\mu})$ is the convex conjugate of $R(\boldsymbol{w})$ and $L_n^*(\boldsymbol{\alpha}_n)$ is the convex conjugate of $L_n(\boldsymbol{\xi}_n)$. The block minimization algorithm of [26] basically performs a dual Block-Coordinate Descent (dual-BCD) over (4) by dividing the whole data set $\mathcal{D}$ into K blocks $\mathcal{D}_{B_1}, ..., \mathcal{D}_{B_K}$, and optimizing a block of dual variables $(\boldsymbol{\alpha}_{B_k}, \boldsymbol{\mu})$ at a time, where $\mathcal{D}_{B_k} = \{(\Phi_n, \boldsymbol{y}_n)\}_{n\in B_k}$ and $\boldsymbol{\alpha}_{B_k} = \{\boldsymbol{\alpha}_n | n \in B_k\}$.

In [26], the dual problem (4) is derived explicitly in order to perform the algorithm. However, for many sparsity-inducing regularizer such as L1-norm and nuclear norm, it is more efficient and convenient to solve (1) in the primal [6, 28]. Therefore, here instead of explicitly forming the dual problem, we express it implicitly as

$$G(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},\boldsymbol{\xi}} \quad \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{\xi}), \tag{5}$$

where $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{\xi})$ is the Lagrangian function of (1), and maximize (5) w.r.t. a block of variables $\boldsymbol{\alpha}_{B_k}$ from the primal instead of dual by strong duality

$$\max_{\alpha_{B_k}} \left\{ \min_{\boldsymbol{w},\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{\xi}) \right\} = \min_{\boldsymbol{w},\boldsymbol{\xi}} \left\{ \max_{\alpha_{B_k}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{\xi}) \right\} \tag{6}$$

with other dual variables $\{\boldsymbol{\alpha}_{B_j} = \boldsymbol{\alpha}_{B_j}^t\}_{j\neq k}$ fixed. The maximization of dual variables $\boldsymbol{\alpha}_{B_k}$ in (6) then enforces the primal equalities $\Phi_n \boldsymbol{w} = \boldsymbol{\xi}_n, \ n \in B_k$, which results in the block minimization problem

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d, \boldsymbol{\xi}_n\in\mathbb{R}^p} \quad & R(\boldsymbol{w}) + \sum_{n\in B_k} L_n(\boldsymbol{\xi}_n) + \boldsymbol{\mu}_{B_k}^{tT} \boldsymbol{w} \\ s.t. \quad & \Phi_n \boldsymbol{w} = \boldsymbol{\xi}_n, \ n \in B_k, \end{aligned} \tag{7}$$

[1] The logistic loss is strongly convex when its input $\boldsymbol{\xi}$ are within a bounded range, which is true as long as we have a non-zero regularizer $R(\boldsymbol{w})$.

where $\boldsymbol{\mu}^t_{B_k} = \sum_{n \notin B_k} \Phi_n^T \boldsymbol{\alpha}_n^t$. Note that, in (7), variables $\{\boldsymbol{\xi}_n\}_{n \notin B_k}$ have been dropped since they are not relevant to the block of dual variables $\boldsymbol{\alpha}_{B_k}$, and thus given the d dimensional vector $\boldsymbol{\mu}^t_{B_k}$, one can solve (7) without accessing data $\{(\Phi_n, \boldsymbol{y}_n)\}_{n \notin B_k}$ outside the block B_k. Throughout the dual-BCD algorithm, we maintain d-dimensional vector $\boldsymbol{\mu}^t = \sum_{n=1}^N \Phi_n^T \boldsymbol{\alpha}_n^t$ and compute $\boldsymbol{\mu}^t_B$ via

$$\boldsymbol{\mu}^t_B = \boldsymbol{\mu}^t - \sum_{n \in B_k} \Phi_n^T \boldsymbol{\alpha}_n^t \tag{8}$$

in the beginning of solving each block subproblem (7). Since subproblem (7) is of the same form to the original problem (1) except for one additional linear augmented term $\boldsymbol{\mu}^T_{B_k} \boldsymbol{w}$, one can adapt the solver of (1) to solve (7) easily by providing an augmented version of the gradient

$$\nabla_{\boldsymbol{w}} \bar{F}(\boldsymbol{w}, \boldsymbol{\xi}) = \nabla_{\boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\xi}) + \boldsymbol{\mu}^t_{B_k}$$

to the solver, where $\bar{F}(.)$ denotes the function with augmented terms and $F(.)$ denotes the function without augmented terms. Note the augmented term $\boldsymbol{\mu}^t_{B_k}$ is constant and separable w.r.t. coordinates, so it adds little overhead to the solver. After obtaining solution $(\boldsymbol{w}^*, \boldsymbol{\xi}^*_{B_k})$ from (7), we can derive the corresponding optimal dual variables $\boldsymbol{\alpha}_{B_k}$ for (6) according to the KKT condition and maintain $\boldsymbol{\mu}$ subsequently by

$$\boldsymbol{\alpha}_n^{t+1} = \nabla_{\boldsymbol{\xi}_n} L_n(\boldsymbol{\xi}_n^*), \; n \in B_k \tag{9}$$

$$\boldsymbol{\mu}^{t+1} = \boldsymbol{\mu}^t_{B_k} + \sum_{n \in B_k} \Phi_n^T \boldsymbol{\alpha}_n^{t+1}. \tag{10}$$

The procedure is summarized in Algorithm 1, which requires a total memory capacity of $O(d + |\mathcal{D}_{B_k}| + p|B_k|)$. The factor d comes from the storage of $\boldsymbol{\mu}^t$, $\boldsymbol{w}^t$, factor $|\mathcal{D}_{B_k}|$ comes from the storage of data block, and the factor $p|B_k|$ comes from the storage of $\boldsymbol{\alpha}_{B_k}$. Note this requires the same space complexity as that required in the original algorithm proposed for linear SVM [26], where $p = 1$ for the binary classification setting.

4 Dual-Augmented Block Minimization

The Block Minimization Algorithm 1, though can be applied to the general regularized ERM problem (1), it is not guaranteed that the sequence $\{\boldsymbol{\alpha}^t\}_{t=0}^\infty$ produced by Algorithm 1 converges to global optimum of (1). In fact, the global convergence of Algorithm 1 only happens for some special cases. One sufficient condition for the global convergence of a *Block-Coordinate* Descent algorithm is that the terms in objective function that are not separable w.r.t. blocks must be *smooth* (Definition 2).

The dual objective function (4) (expressed using only $\boldsymbol{\alpha}$) comprises two terms $R^*(-\sum_{n=1}^N \Phi_n^T \boldsymbol{\alpha}_n) + \sum_{n=1}^N L_n^*(\boldsymbol{\alpha}_n)$, where second term is separable w.r.t. to $\{\boldsymbol{\alpha}_n\}_{n=1}^N$, and thus is also separable w.r.t. $\{\boldsymbol{\alpha}_{B_k}\}_{k=1}^K$, while the first term couples variables $\boldsymbol{\alpha}_{B_1}, ..., \boldsymbol{\alpha}_{B_K}$ involving all the blocks. As a result, if $R^*(-\boldsymbol{\mu})$ is a smooth function according to Definition 2, then Algorithm 1 has global convergence to the optimum. However, the following theorem states this is true only when $R(\boldsymbol{w})$ is strongly convex.

Theorem 1 (Strong/Smooth Duality). *Assume $f(.)$ is closed and convex. Then $f(.)$ is smooth with parameter M if and only if its convex conjugate $f^*(.)$ is strongly convex with parameter $m = \frac{1}{M}$.*

A proof of above theorem can be found in [9]. According to Theorem 1, the Block Minimization Algorithm 1 is not globally convergent if $R(\boldsymbol{w})$ is not strongly convex, which however, is the case for most of regularizers other than the L2-norm $R(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$, as discussed in Section 2.

In this section, we propose a remedy to this problem, which by a Dual-Augmented Lagrangian method (or equivalently, Primal Proximal Point method), creates a dual objective function of desired property that iteratively approaches the original objective (1), and results in fast global convergence of the dual-BCD approach.

Algorithm 1 Dual Block Minimization

```
1. Split data $\mathcal{D}$ into blocks $B_1, B_2, ..., B_K$.
2. Initialize $\boldsymbol{\mu}^0 = \mathbf{0}$.
for $t = 0, 1, ...$ do
    3.1. Draw $k$ uniformly from $[K]$.
    3.2. Load $\mathcal{D}_{B_k}$ and $\boldsymbol{\alpha}^t_{B_k}$ into memory.
    3.3. Compute $\boldsymbol{\mu}^t_{B_k}$ from (8).
    3.4. Solve (7) to obtain $(\boldsymbol{w}^*, \boldsymbol{\xi}^*_{B_k})$.
    3.5. Compute $\boldsymbol{\alpha}^{t+1}_{B_k}$ by (9).
    3.6. Maintain $\boldsymbol{\mu}^{t+1}$ through (10).
    3.7. Save $\boldsymbol{\alpha}^{t+1}_{B_k}$ out of memory.
end for
```

Algorithm 2 Dual-Aug. Block Minimization

```
1. Split data $\mathcal{D}$ into blocks $B_1, B_2, ..., B_K$.
2. Initialize $\boldsymbol{w}^0 = \mathbf{0}$, $\boldsymbol{\mu}^0 = \mathbf{0}$.
for $t = 0, 1, ...$ (outer iteration) do
    for $s = 0, 1, ..., S$ do
        3.1.1. Draw $k$ uniformly from $[K]$.
        3.1.2. Load $\mathcal{D}_{B_k}$, $\boldsymbol{\alpha}^s_{B_k}$ into memory.
        3.1.3. Compute $\boldsymbol{\mu}^s_{B_k}$ from (15).
        3.1.4. Solve (14) to obtain $(\boldsymbol{w}^*, \boldsymbol{\xi}^*_{B_k})$.
        3.1.5. Compute $\boldsymbol{\alpha}^{s+1}_{B_k}$ by (16).
        3.1.6. Maintain $\boldsymbol{\mu}^{s+1}$ through (17).
        3.1.7. Save $\boldsymbol{\alpha}^{s+1}_{B_k}$ out of memory.
    end for
    3.2. $\boldsymbol{w}^{t+1} = \boldsymbol{w}^*(\boldsymbol{\alpha}^S)$.
end for
```

4.1 Algorithm

The *Dual Augmented Lagrangian (DAL)* method (or equivalently, *Proximal Point Method*) modifies the original problem by introducing a sequence of Proximal Maps

$$\boldsymbol{w}^{t+1} = arg\min_{\boldsymbol{w}} \quad F(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{w}^t\|^2, \tag{11}$$

where $F(\boldsymbol{w})$ denotes the ERM problem (1) Under this simple modification, instead of doing Block-Coordinate Descent in the dual of original problem (1), we perform Dual-BCD on the proximal sub-problem (11). As we show in next section, the dual formulation of (11) has the required property for global convergence of the Dual BCD algorithm — all terms involving more than one block of variables $\boldsymbol{\alpha}_{B_k}$ are smooth. Given the current iterate $\boldsymbol{w}^t$, the Dual-Augmented Block Minimization algorithm optimizes the dual of proximal-point problem (11) w.r.t. one block of variables $\boldsymbol{\alpha}_{B_k}$ at a time, keeping others fixed $\{\boldsymbol{\alpha}_{B_j} = \boldsymbol{\alpha}^{(t,s)}_{B_j}\}_{j \neq k}$:

$$\max_{\alpha_{B_k}} \min_{\boldsymbol{w},\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \min_{\boldsymbol{w},\boldsymbol{\xi}} \max_{\alpha_{B_k}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) \tag{12}$$

where $\mathcal{L}(.)$ is the Lagrangian of (11)

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}) = F(\boldsymbol{w}, \boldsymbol{\xi}) + \sum_{n=1}^{N} \boldsymbol{\alpha}_n^T(\Phi_n \boldsymbol{w} - \boldsymbol{\xi}_n) + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{w}^t\|^2. \tag{13}$$

Once again, the maximization w.r.t. $\boldsymbol{\alpha}_{B_k}$ in (12) enforces the equalities $\Phi_n \boldsymbol{w} = \boldsymbol{\xi}_n,\ n \in B_k$ and thus leads to a primal sub-problem involving only data in block B_k:

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, \boldsymbol{\xi}_n \in \mathbb{R}^p} \quad & R(\boldsymbol{w}) + \sum_{n \in B_k} L_n(\boldsymbol{\xi}_n) + \boldsymbol{\mu}^{(t,s)T}_{B_k} \boldsymbol{w} + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 \\ s.t. \quad & \Phi_n \boldsymbol{w} = \boldsymbol{\xi}_n,\ n \in B_k, \end{aligned} \tag{14}$$

where $\boldsymbol{\mu}^{(t,s)}_{B_k} = \sum_{n \notin B_k} \Phi_n^T \boldsymbol{\alpha}_n^{(t,s)}$. Note that (14) is almost the same as (7) except that it has a proximal-point augmented term. Therefore, one can follow the same procedure as in Algorithm 1 to maintain the vector $\boldsymbol{\mu}^{(t,s)} = \sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n^{(t,s)}$ and computes

$$\boldsymbol{\mu}^{(t,s)}_{B_k} = \boldsymbol{\mu}^{(t,s)} - \sum_{n \in B_k} \Phi_n^T \boldsymbol{\alpha}_n^{(t,s)} \tag{15}$$

before solving each block subproblem (14). After obtaining solution $(\boldsymbol{w}^*, \boldsymbol{\xi}^*_{B_k})$ from (14), we update dual variables $\boldsymbol{\alpha}_{B_k}$ as

$$\boldsymbol{\alpha}_n^{(t,s+1)} = \nabla_{\boldsymbol{\xi}_n} L_n(\boldsymbol{\xi}_n^*),\ n \in B_k. \tag{16}$$

and maintain $\boldsymbol{\mu}$ subsequently as

$$\boldsymbol{\mu}^{(t,s+1)} = \boldsymbol{\mu}^{(t,s)}_{B_k} + \sum_{n \in B_k} \Phi_n^T \boldsymbol{\alpha}_n^{(t,s+1)}. \tag{17}$$

The sub-problem (14) is of similar form to the original ERM problem (1). Since the augmented term is a simple quadratic function separable w.r.t. each coordinate, given a solver for (1) working in sufficient-memory condition, one can easily adapt it by modifying

$$\begin{aligned}\nabla_{\boldsymbol{w}}\bar{F}(\boldsymbol{w},\boldsymbol{\xi}) &= \nabla_{\boldsymbol{w}}F(\boldsymbol{w},\boldsymbol{\xi}) + \boldsymbol{\mu}^t_{B_k} + (\boldsymbol{w}-\boldsymbol{w}_t)/\eta_t \\ \nabla^2_{\boldsymbol{w}}\bar{F}(\boldsymbol{w},\boldsymbol{\xi}) &= \nabla^2_{\boldsymbol{w}}F(\boldsymbol{w},\boldsymbol{\xi}) + I/\eta_t,\end{aligned}$$

where $\bar{F}(.)$ denotes the function with augmented terms and $F(.)$ denotes the function without augmented terms. The Block Minimization procedure is repeated until every sub-problem (14) reaches a tolerance ϵ_{in}. Then the proximal point method update $\boldsymbol{w}^{t+1} = \boldsymbol{w}^*(\boldsymbol{\alpha}^{(t,s)})$ is performed, where $\boldsymbol{w}^*(\boldsymbol{\alpha}^{(t,s)})$ is the solution of (14) for the latest dual iterate $\boldsymbol{\alpha}^{(t,s)}$. The resulting algorithm is summarized in Algorithm 2.

4.2 Analysis

In this section, we analyze the convergence rate of Algorithm 2 to the optimum of (1). First, we show that the proximal-point formulation (11) has a dual problem with desired property for the global convergence of Block-Coordinate Descent. In particular, since the dual of (11) takes the form

$$\min_{\boldsymbol{\alpha}_n\in\mathbb{R}^p} \quad \tilde{R}^*(-\sum_{n=1}^{N}\Phi_n^T\boldsymbol{\alpha}_n) + \sum_{n=1}^{N} L_n^*(\boldsymbol{\alpha}_n) \tag{18}$$

where $\tilde{R}^*(.)$ is the convex conjugate of $\tilde{R}(\boldsymbol{w}) = R(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w}-\boldsymbol{w}_t\|^2$, and since $\tilde{R}(\boldsymbol{w})$ is strongly convex with parameter $m = 1/\eta_t$, the convex conjugate $\tilde{R}^*(.)$ is smooth with parameter $M = \eta_t$ according to Theorem 1. Therefore, (18) is in the composite form of a convex, smooth function plus a convex, block-separable function. This type of function has been widely studied in the literature of Block-Coordinate Descent [13]. In particular, one can show that a Block-Coordinate Descent applied on (18) has global convergence to optimum with a fast rate by the following theorem.

Theorem 2 (BCD Convergence). *Let the sequence $\{\boldsymbol{\alpha}^s\}_{s=1}^{\infty}$ be the iterates produced by Block Coordinate Descent in the inner loop of Algorithm 2, and K be the number of blocks. Denote $\tilde{F}^*(\boldsymbol{\alpha})$ as the dual objective function of* (18) *and $\tilde{F}^*_{opt}$ the optimal value of* (18). *Then with probability $1-\rho$,*

$$\tilde{F}^*(\boldsymbol{\alpha}^s) - \tilde{F}^*_{opt} \leq \epsilon, \;\; for \;\; s \geq \beta K \log(\frac{\tilde{F}^*(\boldsymbol{\alpha}^0) - \tilde{F}^*_{opt}}{\rho\epsilon}) \tag{19}$$

for some constant $\beta > 0$ if (i) $L_n(.)$ is smooth, or (ii) $L_n(.)$ is polyhedral function and $R(.)$ is also polyhedral or smooth. Otherwise, for any convex $L_n(.)$, $R(.)$ we have

$$\tilde{F}^*(\boldsymbol{\alpha}^s) - \tilde{F}^*_{opt} \leq \epsilon, \;\; for \;\; s \geq \frac{cK}{\epsilon} \log(\frac{\tilde{F}^*(\boldsymbol{\alpha}^0) - \tilde{F}^*_{opt}}{\rho\epsilon}) \tag{20}$$

for some constant $c > 0$.

Note the above analysis (in appendix) does not assume exact solution of each block subproblem. Instead, it only assumes each block minimization step leads to a dual ascent amount proportional to that produced by a single (dual) proximal gradient ascent step on the block of dual variables. For the outer loop of Primal Proximal-Point (or Dual Augmented Lagrangian) iterates (11), we show the following convergence theorem.

Theorem 3 (Proximal Point Convergence). *Let $F(\boldsymbol{w})$ be objective of the regularized ERM problem* (1), *and $\mathcal{R} = \max_{\boldsymbol{v}}\max_{\boldsymbol{w}}\{\|\boldsymbol{v}-\boldsymbol{w}\| : F(\boldsymbol{w}) \leq F(\boldsymbol{w}^0), F(\boldsymbol{v}) \leq F(\boldsymbol{w}^0)\}$ be the radius of initial level set. The sequence $\{\boldsymbol{w}^t\}_{t=1}^{\infty}$ produced by the Proximal-Point update* (11) *with $\eta_t = \eta$ has*

$$F(\boldsymbol{w}^{t+1}) - F_{opt} \leq \epsilon, \;\; for \;\; t \geq \tau \log(\frac{\omega}{\epsilon}). \tag{21}$$

for some constant $\tau, \omega > 0$ if both $L_n(.)$ and $R(.)$ are (i) strictly convex and smooth or (ii) polyhedral. Otherwise, for any convex $F(\boldsymbol{w})$ we have

$$F(\boldsymbol{w}^{t+1}) - F_{opt} \leq \mathcal{R}^2/(2\eta t).$$

The following theorem further shows that solving sub-problem (11) inexactly with tolerance ϵ/t suffices for convergence to ϵ overall precision, where t is the number of outer iterations required.

Theorem 4 (Inexact Proximal Map)**.** *Suppose, for a given dual iterate $\boldsymbol{w}^t$, each sub-problem* (11) *is solved inexactly s.t. the solution $\hat{\boldsymbol{w}}^{t+1}$ has*

$$\|\hat{\boldsymbol{w}}^{t+1} - \mathbf{prox}_{\eta_t F}(\boldsymbol{w}^t)\| \leq \epsilon_0. \tag{22}$$

Then let $\{\hat{\boldsymbol{w}}^t\}_{t=1}^{\infty}$ be the sequence of iterates produced by inexact proximal updates and $\{\boldsymbol{w}^t\}_{t=1}^{\infty}$ as that generated by exact updates. After t *iterations, we have*

$$\|\hat{\boldsymbol{w}}^t - \boldsymbol{w}^t\| \leq t\epsilon_0. \tag{23}$$

Note for $L_n(.)$, $R(.)$ being strictly convex and smooth, or polyhedral, t is of order $O(\log(1/\epsilon))$, and thus it only requires $O(K\log(1/\epsilon)\log(t/\epsilon)) = O(K\log^2(1/\epsilon))$ overall number of block minimization steps to achieve ϵ suboptimality. Otherwise, as long as $L_n(.)$ is smooth, for any convex regularizer $R(.)$, t is of order $O(1/\epsilon)$, so it requires $O(K(1/\epsilon)\log(t/\epsilon)) = O(\frac{K\log(1/\epsilon)}{\epsilon})$ total number of block minimization steps.

4.3 Practical Issues

4.3.1 Solving Sub-Problem Inexactly

While the analysis in Section 4.2 assumes exact solution of subproblems, in practice, the Block Minimization framework does not require solving subproblem (11), (14) exactly. In our experiments, it suffices for the fast convergence of proximal-point update (11) to solve subproblem (14) for only a single pass of all blocks of variables $\boldsymbol{\alpha}_{B_1}$,..., $\boldsymbol{\alpha}_{B_K}$, and limit the number of iterations the designated solver spends on each subproblem (7), (14) to be no more than some parameter T_{max}.

4.3.2 Random Selection w/o Replacement

In Algorithm 1 and 2, the block to be optimized is chosen uniformly at random from $k \in \{1, ..., K\}$, which eases the analysis for proving a better convergence rate [13]. However, in practice, to avoid unbalanced update frequency among blocks, we do *random sampling without replacement* for both Algorithm 1 and 2, that is, for every K iterations, we generate a random permutation $\pi_1, ..., \pi_K$ of block index $1, .., K$ and optimize block subproblems (7), (14) according to the order $\pi_1, .., \pi_K$. This also eases the checking of inner-loop stopping condition.

4.3.3 Storage of Dual Variables

Both the algorithms 1 and 2 need to store the dual variables $\boldsymbol{\alpha}_{B_k}$ into memory and load/save them from/to some secondary storage units, which requires a time linear to $p|B_k|$. For some problems, such as multi-label classification with large number of labels or structured prediction with large number of factors, this can be very expensive. In this situation, one can instead maintain $\boldsymbol{\mu}_{\bar{B}_k} = \sum_{n\in B_k} \Phi_n^T \alpha_n = \boldsymbol{\mu} - \boldsymbol{\mu}_{B_k}$ directly. Note $\boldsymbol{\mu}_{\bar{B}_k}$ has I/O and storage cost linear to d, which can be much smaller than $p|B_k|$ in a low-dimensional problem.

5 Experiment

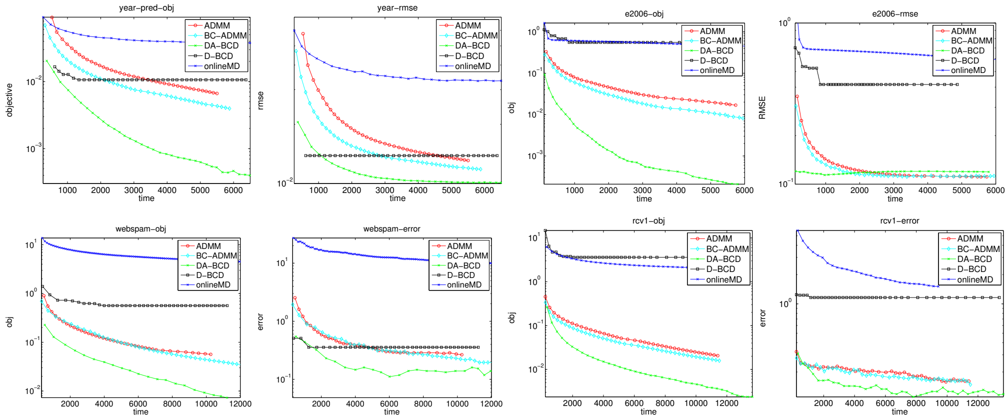
In this section, we compare the proposed *Dual Augmented Block Minimization* framework (Algorithm 2) to the vanilla Dual Block Coordinate Descent algorithm [26] and methods adopted from Online and Distributed Learning. The experiments are conducted on the problem of $L1$-regularized L2-loss SVM [27] and the (*Lasso*) (L1-regularized Regression) problem [17] in the limited-memory setting with data size 10 times larger than the available memory. For both problems, we use state-of-the-art randomized coordinate descent method [13, 27] as the solver for solving sub-problems (7), (14), (59), (63), and we set parameter $\eta_t = 1$, $\lambda = 1$ (of L1-regularizer) for all experiments. Four public benchmark data sets are used— *webspam*, *rcv1-binary* for classification and *year-pred*, *E2006* for regression, which can be obtained from the LIBSVM data set collections. For *year-pred* and *E2006*, the features are generated from Random Fourier Features [12, 23] that approximate the effect of Gaussian RBF kernel. Table 1 summarizes the data statistics. The algorithms in comparison and their shorthands are listed below, where all solvers are implemented in C/C++ and run on 64-bit machine with 2.83GHz Intel(R) Xeon(R) CPU. We constrained the process to use no more than $1/10$ of memory required to store the whole data.

- **OnlineMD**: Stochastic Mirror Descent method specially designed for L1-regularized problem proposed in [15] with step size chosen from 10^{-2}-10^{2} for best performance.

Table 1: Data Statistics: Summary of data statistics when stored using sparse format. The last two columns specify memory consumption in (MB) of the whole data and that of a block when data is split into $K = 10$ partitions.

Data	#train	#test	dimension	#non-zeros	Memory	Block
webspam	315,000	31,500	680,714	1,174,704,031	20,679	2,068
rcv1	202,420	20,242	7,951,176	656,977,694	12,009	1,201
year-pred	463,715	51,630	2,000	927,893,715	13,702	1,370
E2006	16,087	3,308	30,000	8,088,636	8,088	809

Figure 1: Relative function value difference to the optimum and Testing RMSE (Accuracy) on LASSO (top) and L1-regularized L2-SVM (bottom). (RMSE best for year-pred: 9.1320; for E2006: 0.4430), (Accuracy best for for webspam: 0.4761%; best for rcv1: 2.213%).



- **D-BCD**[2]: Dual Block-Coordinate Descent method (Algorithm 1).
- **DA-BCD**: Dual-Augmented Block Minimization (Algorithm 2).
- **ADMM**: ADMM for limited-memory learning (Algorithm 3 in appendix-B).
- **BC-ADMM**: Block-Coordinate ADMM that updates a randomly chosen block of dual variables at a time for limited-memory learning (Algorithm 4 in appendix-B) .

We use *wall clock time* that includes both I/O and computation as measure for training time in all experiments. In Figure 5, three measures are plotted versus the training time: Relative objective function difference to the optimum, Testing RMSE and Accuracy. Figure 5 shows the results, where as expected, the dual Block Coordinate Descent (D-BCD) method without augmentation cannot improve the objective after certain number of iterations. However, with extremely simple modification, the Dual-Augmented Block Minimization (DA-BCD) algorithm becomes not only globally convergent but with a rate several times faster than other approaches. Among all methods, the convergence of *Online Mirror Descent* (SMIDAS) is significantly slower, which is expected since (i) the online Mirror Descent on a non-smooth, non-strongly convex function converges at a rate qualitatively slower than the linear convergence rate of DA-BCD and ADMM [15, 16], and (ii) Online method does not utilize the available memory capacity and thus spends unbalanced time on I/O and computation. For methods adopted from distributed optimization, the experiment shows BC-ADMM consistently, but only slightly, improves ADMM, and both of them converge much slower than the DA-BCD approach, presumably due to the conservative updates on the dual variables.

Acknowledgement We thank to the support of Telecommunication Lab., Chunghwa Telecom Co., Ltd via TL-103-8201, AOARD via No. FA2386-13-1-4045, Ministry of Science and Technology, National Taiwan University and Intel Co. via MOST102-2911-I-002-001, NTU103R7501, 102-2923-E-002-007-MY2, 102-2221-E-002-170, 103-2221-E-002-104-MY2.

[2] The objective value obtained from D-BCD fluctuates a lot, in figures we plot the lowest values achieved by D-BCD from the beginning to time t.

References

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

[2] K. Chang and D. Roth. Selective block minimization for faster convergence of limited memory large-scale linear models. In *SIGKDD*. ACM, 2011.

[3] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] A. Hoffman. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standards*, 1952.

[5] M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers, 2012.

[6] C. Hsieh, I. Dhillon, P. Ravikumar, S. Becker, and P. Olsen. Quic & dirty: A quadratic approximation approach for dirty statistical models. In *NIPS*, 2014.

[7] M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. Jordan. Communication-efficient distributed dual coordinate ascent. In *NIPS*, 2014.

[8] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, 2005.

[9] S. Kakade, S. Shalev-Shwartz, and A. Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. *CoRR*, 2009.

[10] C. Ma, V. Smith, M. Jaggi, M. Jordan, P. Richtárik, and M. Takáč. Adding vs. averaging in distributed primal-dual optimization. *ICML*, 2015.

[11] G. Obozinski, L. Jacob, and J. Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint*, 2011.

[12] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[13] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2014.

[14] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 2011.

[15] S. Shalev-Shwartz and A. Tewari. Stochastic methods for l1-regularized loss minimization. *JMLR*, 2011.

[16] N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. In *NIPS*, 2011.

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1996.

[18] R. Tomioka, T. Suzuki, and M. Sugiyama. Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *JMLR*, 2011.

[19] I. Trofimov and A. Genkin. Distributed coordinate descent for l1-regularized logistic regression. *arXiv preprint*, 2014.

[20] P. Wang and C. Lin. Iteration complexity of feasible descent methods for convex optimization. *JMLR*, 2014.

[21] I. Yen, C. Chang, T. Lin, S., and S. Lin. Indexed block coordinate descent for large-scale linear classification with limited memory. In *SIGKDD*. ACM, 2013.

[22] I. Yen, C. Hsieh, P. Ravikumar, and I. Dhillon. Constant nullspace strong convexity and fast convergence of proximal methods under high-dimensional settings. In *NIPS*, 2014.

[23] I. Yen, T. Lin, S. Lin, P. Ravikumar, and I. Dhillon. Sparse random feature algorithm as coordinate descent in hilbert space. In *NIPS*, 2014.

[24] I. Yen, X. Lin, K. Zhong, P. Ravikumar, and I. Dhillon. A convex exemplar-based approach to MAD-Bayes dirichlet process mixture models. In *ICML*, 2015.

[25] I. Yen, K. Zhong, C. Hsieh, P. Ravikumar, and I. Dhillon. Sparse linear programming via primal and dual augmented coordinate descent. In *NIPS*, 2015.

[26] H. Yu, C. Hsieh, . Chang, and C. Lin. Large linear classification when data cannot fit in memory. *SIGKDD*, 2010.

[27] G. Yuan, K. Chang, C. Hsieh, and C. Lin. A comparison of optimization methods and software for large-scale L1-regularized linear classification. *JMLR*, 2010.

[28] K. Zhong, I. Yen, I. Dhillon, and P. Ravikumar. Proximal quasi-Newton for computationally intensive l1-regularized m-estimators. In *NIPS*, 2014.

6 Appendix-A. Convergence Analysis

6.1 Convergence of Randomized Block Coordinate Descent

We first establish the linear convergence of Randomized Block Coordinate Descent (RBCD) when $L_n(.)$ is smooth in the sense that its first derivative $L'_n(.)$ is Lipschitz-continuous with parameter M_L, which then implies $L_n^*(\boldsymbol{\alpha}_n)$ is strongly convex with parameter $1/M_L$.

Theorem 2-1 (Dual-RBCD for Smooth Loss). *Let the sequence $\{\boldsymbol{\alpha}^s\}_{s=1}^{\infty}$ be the iterates produced by RBCD in the inner loop of Algorithm 2, and K be the number of blocks. Denote $\tilde{F}^*(\boldsymbol{\alpha})$ as the dual objective function of* (18) *and $\tilde{F}^*_{opt}$ the optimal function value of* (18)*. Then with probability $1-\rho$,*

$$\tilde{F}^*(\boldsymbol{\alpha}^s) - \tilde{F}^*_{opt} \leq \epsilon, \;\; for \;\; s \geq \frac{K}{1-c_1}\log(\frac{\tilde{F}^*(\boldsymbol{\alpha}^0) - \tilde{F}^*_{opt}}{\rho\epsilon}) \tag{24}$$

if $L_n(.)$ is smooth, where $0 < c_1 < 1$ is a constant depends on the smoothness parameter of $L_n(.)$.

Proof. This is a special case of theorem 6 and theorem 4 in [13], where they consider composite objective function of the form

$$F(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}) + \Psi(\boldsymbol{\alpha}), \tag{25}$$

where $f(\boldsymbol{\alpha})$ is a convex, smooth function, and $\Psi(\boldsymbol{\alpha})$ is a convex, block-separable function. In our case,

$$f(\boldsymbol{\alpha}) = \tilde{R}^*(-\sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n), \;\; \Psi(\boldsymbol{\alpha}) = \sum_{n=1}^{N} L_n^*(\boldsymbol{\alpha}_n). \tag{26}$$

Note $\tilde{R}^*(.)$ is smooth w.r.t. $\boldsymbol{\alpha}_{B_k}$ with parameter $M_R = \eta_t B^2$, where $B \geq \|\Phi_{B_k}\|_2$ is an upper bound on the ℓ_2-norm of each block's design matrix. If the loss function $L_n(.)$ is smooth with paramter M_L, by Theorem 1, $\Psi(\boldsymbol{\alpha})$ is strongly convex with parameter $1/M_L$, and thus, based on [theorem 6, 21], (24) holds with

$$c_1 = \left\{ \begin{array}{ll} 1 - \frac{1}{4M_R M_L} & , \, if M_R M_L \geq \frac{1}{2} \\ M_R M_L & , \, o.w.. \end{array} \right. \tag{27}$$

□

For some important classes of ERM, such as Support Vector Machine (SVM) and its variants (e.g. Multiclass, Structral SVM), $L_n(\boldsymbol{\alpha}_n)$ is not smooth but piecewise-linear. In the following, we show that the linear convergence of RBCD holds for any loss $L_n(\boldsymbol{\alpha}_n)$ with polyhedral epigraph if $R(\boldsymbol{w})$ is also polyhedral or smooth. The proof utilizes a restricted version of Strong Convexity called Constant Nullspace Strong Convexity [20, 22] and obtains a much tighter bound for RBCD than the bound proved in [20] for general feasible descent method. The proof follows is a generalization of that in [25] for proving linear convergence of RCD applied to the Augmented Lagrangian of Linear Program.

The augmented dual objective function (25), after some algebraic rearrangement, is equivalent to

$$\min_{\boldsymbol{\alpha},\boldsymbol{\mu}} \;\; \sum_{n=1}^{N} L_n^*(\boldsymbol{\alpha}_n) + R^*(-\boldsymbol{\mu}) + \frac{\eta_t}{2}\|\sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n - \boldsymbol{\mu} + \boldsymbol{w}^t/\eta_t\|^2 \tag{28}$$

up to a constant. For $L_n^*(\boldsymbol{\alpha}_n)$, $R^*(-\boldsymbol{\mu})$ being polyhedral, their epigraphs $\mathbf{epi}(L_n)$, $\mathbf{epi}(R)$ are polyhedrons and thus (28) can be also formulated as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{t},s} & \sum\limits_{n=1}^{N} t_n + r + \frac{\eta_t}{2}\|\sum\limits_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n - \boldsymbol{\mu} + \boldsymbol{w}^t/\eta_t\|^2 \\ s.t. & (\boldsymbol{\alpha}_n, t_n) \in \mathbf{epi}(L_n) \\ & (\boldsymbol{\mu}, r) \in \mathbf{epi}(R), \end{array} \tag{29}$$

which is of the form

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & F(\boldsymbol{x}) = g(\bar{\Phi}^T \boldsymbol{x}) + \boldsymbol{c}^T \boldsymbol{x} \\ s.t. & \boldsymbol{x} \in \mathcal{P} \end{array} \tag{30}$$

where $g(\boldsymbol{z}) = \frac{\eta_t}{2}\|\boldsymbol{z} + \boldsymbol{w}^t/\eta_t\|^2$ is a strongly convex function, $\mathcal{P}$ is a polyhedral set

$$\mathcal{P} = \{(\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{\mu}, r) \mid (\boldsymbol{\alpha}_n, t_n) \in \mathbf{epi}(L_n), (\boldsymbol{\mu}, r) \in \mathbf{epi}(R)\},$$

and

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{t} \\ \boldsymbol{\mu} \\ r \end{bmatrix} \qquad \bar{\Phi} = \begin{bmatrix} \Phi \\ O_{N,d} \\ -I_{d,d} \\ O_{1,d} \end{bmatrix} \qquad \boldsymbol{c} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ 1 \end{bmatrix}.$$

We will use $I_{\boldsymbol{\alpha}}$, $I_{\boldsymbol{t}}$, $I_{\boldsymbol{\mu}}$ and I_s denote the set of variable indexes j that correspond to $\boldsymbol{\alpha}$, $\boldsymbol{t}$, $\boldsymbol{\mu}$ and r respectively. For this type of objective function, we can show that the set of optimal solutions is a polyhedron defined by the following Lemma.

Lemma 1 (Lemma 4.2 of [20])**.** *The optimal solutions to problem* (30) *forms a polyhedral set*

$$\mathcal{S} = \{\boldsymbol{x} \mid \bar{\Phi}^T\boldsymbol{x} = \boldsymbol{p}^*,\ \boldsymbol{c}^T\boldsymbol{x} = q^*,\ \boldsymbol{x} \in \mathcal{P}\} \tag{31}$$

for some unique $\boldsymbol{p}^*$, q^*.

Furthermore, we can utilize the Hoffman's bound (defined in the following) to bound the distance of any point $\boldsymbol{x}$ to the optimal solution set $\mathcal{S}$.

Lemma 2 (Hoffman's Bound)**.** *Let $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^d \mid A\boldsymbol{x} \leq \boldsymbol{c},\ E\boldsymbol{x} = \boldsymbol{c}\}$ be a polyhedral set. Then for any point $\boldsymbol{x} \in \mathbb{R}^d$,*

$$\|\boldsymbol{x} - \Pi_{\mathcal{S}}(\boldsymbol{x})\|_2^2 \leq \theta \left\| \begin{array}{c} [A\boldsymbol{x} - \boldsymbol{c}]_+ \\ E\boldsymbol{x} - \boldsymbol{c} \end{array} \right\|_2^2 \tag{32}$$

where $\Pi_{\mathcal{S}}(\boldsymbol{x}) = arg\min_{\boldsymbol{y}\in\mathcal{S}}\ \|\boldsymbol{y} - \boldsymbol{x}\|$ is the projection of $\boldsymbol{x}$ to the set $\mathcal{S}$, and $\theta > 0$ is a constant depending on the polyhedral set $\mathcal{S}$.

Proof. The Hoffman's bound first appears in [4] and a proof for the ℓ_2-norm's version (32) and the definition of the constant $\theta(\mathcal{S})$ can be found in [20] (lemma 4.3). □

By Lemma 2, for any feasible $\boldsymbol{x} \in \mathcal{P}$, we obtain error bound

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \leq \theta(\mathcal{S})\left(\|\bar{\Phi}^T\boldsymbol{x} - \boldsymbol{p}^*\|^2 + \|\boldsymbol{c}^T\boldsymbol{x} - q^*\|^2\right), \tag{33}$$

which plays a crucial role in the proof of linear convergence.

The RBCD algorithm performed on (25) can be considered as minimizing (29) w.r.t. a block of dual variables $\{(\boldsymbol{\alpha}_n, t_n)\}_{n\in B_k}$ together with $(\boldsymbol{\mu}, s)$, while fixing all other variables $\{(\boldsymbol{\alpha}_n, t_n)\}_{n\notin B_k}$. In the following, we show that each block minimization step leads to a significant progress.

Lemma 3 (Descent Amount)**.** *The expected descent amount for each Block Minimization step of Algorithm 2 has*

$$\mathbb{E}[F(\boldsymbol{x}^{k+1})] - F(\boldsymbol{x}^k) \leq \frac{1}{K}\left(\min_{\boldsymbol{\delta}}\ h(\boldsymbol{x}^k + \boldsymbol{\delta}) + \langle\nabla F(\boldsymbol{x}^k), \boldsymbol{\delta}\rangle + \frac{M\eta_t}{2}\|\boldsymbol{\delta}\|^2\right), \tag{34}$$

where

$$h(\boldsymbol{x}) = \left\{ \begin{array}{ll} 0, & \boldsymbol{x} \in \mathcal{P} \\ \infty, & o.w. \end{array} \right. \tag{35}$$

and $M \geq \max_{k\in[K]} \|\Phi_{B_k}\|_2^2$ denotes a bound on the spectral norm of each block's design matrix.

Proof. First, notice that RBCD optimizes the function form of only variable $\boldsymbol{\alpha}$ while maintains other variables $(\boldsymbol{t}, \boldsymbol{\mu}, s)$ as their optimal values in each block minimization step, so we have

$$\mathbf{0} = \min_{\boldsymbol{\mu}, r}\ h(\boldsymbol{x}) + \nabla F(\boldsymbol{x}^s)^T(\boldsymbol{x} - \boldsymbol{x}^s) + \frac{M\eta_t}{2}\|\boldsymbol{x} - \boldsymbol{x}^s\|^2. \tag{36}$$

The algorithm picks coordinate uniformly from $\{(\boldsymbol{\alpha}_{B_k}, \boldsymbol{t}_{B_k})\}_{k=1}^K$ to update. Since the constant M upper bounds $\|\nabla_{\boldsymbol{\alpha}_{B_k}, \boldsymbol{t}_{B_k}} F(\boldsymbol{x})\|_2^2$, we have

$$\begin{aligned} F(\boldsymbol{x}^{s+1}) - F(\boldsymbol{x}^s) &= F(\boldsymbol{\alpha}^{s+1}, t^*(\boldsymbol{\alpha}^{s+1}), \boldsymbol{\mu}^*(\boldsymbol{\alpha}^{s+1}), r^*(\boldsymbol{\alpha}^{s+1})) - F(\boldsymbol{x}^s) \\ &\leq F(\boldsymbol{\alpha}^{s+1}, t^*(\boldsymbol{\alpha}^{s+1}), \boldsymbol{\mu}^s, r^s) - F(\boldsymbol{x}^s) \\ &\leq \min_{\boldsymbol{\delta}_{B_k}}\ h(\boldsymbol{x}^s + \boldsymbol{\delta}_{B_k}) + \nabla_{B_k} F(x^k)^T\boldsymbol{\delta}_{B_k} + \frac{M\eta_t}{2}\|\boldsymbol{\delta}_{B_k}\|^2. \end{aligned}$$

where $\boldsymbol{\delta}_{B_k}$ denotes a change of variables restricted on $(\Delta\boldsymbol{\alpha}_{B_k}, \Delta\boldsymbol{t}_{B_k})$ with all other variables fixed. Note the minimization in (69) is seperable w.r.t $\{\boldsymbol{\delta}_{B_k}\}_{k=1}^{K}$. Therefore, taking expectation of LHS and RHS w.r.t. k yields the result. □

Before moving on, note that function $g(\boldsymbol{z}) = \frac{\eta_t}{2}\|\boldsymbol{z} + \boldsymbol{w}^t/\eta_t\|^2$ is locally Lipschitz-continuous with constant $L_g = \eta_t R_z$ for $\boldsymbol{z}$ satisfying $\|\boldsymbol{z} + \boldsymbol{w}^t/\eta_t\| \leq R_z$, that is,

$$|g(\boldsymbol{z}_1) - g(\boldsymbol{z}_2)| \leq L_g\|\boldsymbol{z}_1 - \boldsymbol{z}_2\| \tag{37}$$

for $\forall \boldsymbol{z}_1, \boldsymbol{z}_2$ with $\|\boldsymbol{z}_1 + \boldsymbol{w}^t/\eta_t\| \leq R_z$, $\|\boldsymbol{z}_2 + \boldsymbol{w}^t/\eta_t\| \leq R_z$, where R_z is an upper bound on the magnitude of iterates $\|\boldsymbol{w}^{t+1}\|/\eta_t = \|\bar{\Phi}^T\boldsymbol{x}^t + \boldsymbol{w}^t/\eta_t\|$.

From simplicity of analysis, in the following, we slightly loosen upper bounds by setting constants $L_g \leftarrow \max(L_g, 1)$, $M \leftarrow \max(M, 1)$, $\theta \leftarrow \max(\theta, 1)$, such that $L_g, M, \theta \geq 1$. Then we are ready to prove the main theorem of this section.

Theorem 5 (Linear Convergence)**.** *The iterates $\{\boldsymbol{x}^s\}_{s=0}^{\infty}$ of Block Minimization for polyhedral $L_n(.)$, $R(.)$ satisfy*

$$\mathbb{E}[F(\boldsymbol{x}^{s+1})] - F^* \leq \left(1 - \frac{1}{K\gamma}\right)(\mathbb{E}[F(\boldsymbol{x}^s)] - F^*)$$

where F^* is the optimum of (28) *and*

$$\gamma = \max\left\{16\eta_t M\theta(F^0 - F^*)\ ,\ 2M\theta(1 + 4L_g^2)\ ,\ 6\right\}.$$

Proof. Let $\boldsymbol{x}^* = \Pi_{\mathcal{S}}(\boldsymbol{x}^s)$ be the projection of $\boldsymbol{x}^s$ to the set of optimal solutions. From Lemma 3, we have

$$\begin{aligned}
\mathbb{E}[F(\boldsymbol{x}^{s+1})] - F(\boldsymbol{x}^s) &\leq \frac{1}{K}\left(\min_{\boldsymbol{\delta}}\ h(\boldsymbol{x}^s + \boldsymbol{\delta}) + \langle\nabla F(\boldsymbol{x}^s), \boldsymbol{\delta}\rangle + \frac{M\eta_t}{2}\|\boldsymbol{\delta}\|^2\right) \\
&\leq \frac{1}{K}\left(\min_{\boldsymbol{\delta}}\ h(\boldsymbol{x}^s + \boldsymbol{\delta}) + F(\boldsymbol{x}^s + \boldsymbol{\delta}) - F(\boldsymbol{x}^s) + \frac{M\eta_t}{2}\|\boldsymbol{\delta}\|^2\right) \\
&\leq \frac{1}{K}\left(\min_{a\in[0,1]}\ F(\boldsymbol{x}^s + a(\boldsymbol{x}^* - \boldsymbol{x}^s)) - F(\boldsymbol{x}^s) + \frac{M\eta_t a^2}{2}\|\boldsymbol{x}^* - \boldsymbol{x}^s\|^2\right) \\
&\leq \frac{1}{K}\left(\min_{a\in[0,1]}\ -a(F(\boldsymbol{x}^s) - F(\boldsymbol{x}^*)) + \frac{M\eta_t a^2}{2}\|\boldsymbol{x}^* - \boldsymbol{x}^s\|^2\right),
\end{aligned} \tag{38}$$

where the second and fourth inequality follow from the convexity of $F(\boldsymbol{x})$, and the third inequality follows from the fact that both $\boldsymbol{x}^*$ and $\boldsymbol{x}^s$ are feasible ($h(\boldsymbol{x}^*) = h(\boldsymbol{x}^s) = 0$). Now based on the error bound inequality (68), we discuss two cases.

Case 1: $4L_g^2\|\bar{\Phi}^T\boldsymbol{x} - \boldsymbol{p}^*\|^2 < (\boldsymbol{c}^T\boldsymbol{x} - q^*)^2$.

In this case, we have

$$\begin{aligned}
\|\boldsymbol{x}^s - \boldsymbol{x}^*\|^2 &\leq \theta\left(\|\bar{\Phi}^T\boldsymbol{x}^s - \boldsymbol{p}^*\|^2 + \|\boldsymbol{c}^T\boldsymbol{x}^s - q^*\|^2\right) \\
&\leq \theta\left(\frac{1}{4L_g^2} + 1\right)(\boldsymbol{c}^T\boldsymbol{x}^s - q^*)^2 \leq 2\theta(\boldsymbol{c}^T\boldsymbol{x}^s - q^*)^2
\end{aligned} \tag{39}$$

and

$$|\boldsymbol{c}^T\boldsymbol{x}^s - q^*| \geq 2L_g\|\bar{\Phi}^T\boldsymbol{x}^s - \boldsymbol{p}^*\| \geq 2|g(\bar{\Phi}^T\boldsymbol{x}^s) - g(\boldsymbol{p}^*)|.$$

Note in this case, $\boldsymbol{c}^T\boldsymbol{x}^s - q^*$ must be non-negative. Otherwise,

$$\begin{aligned}
F(\boldsymbol{x}^s) - F^* &= g(\bar{\Phi}^T\boldsymbol{x}^s) - g(\boldsymbol{p}^*) + (\boldsymbol{c}^T\boldsymbol{x}^s - q^*) \\
&\leq |g(\bar{\Phi}^T\boldsymbol{x}^s) - g(\boldsymbol{p}^*)| - |\boldsymbol{c}^T\boldsymbol{x}^s - q^*| \\
&\leq -\frac{1}{2}|\boldsymbol{c}^T\boldsymbol{x}^s - q^*| < 0,
\end{aligned}$$

leads to contradiction (since $\boldsymbol{x}^s$ is feasible, $F(\boldsymbol{x}^s)$ cannot be smaller than F^*). Therefore, we have

$$\begin{aligned} F(\boldsymbol{x}^s) - F^* &= g(\bar{\Phi}^T \boldsymbol{x}^s) - g(\boldsymbol{p}^*) + \boldsymbol{c}^T \boldsymbol{x}^s - q^* \\ &\geq -|g(\bar{\Phi}^T \boldsymbol{x}^s) - g(\boldsymbol{p}^*)| + \boldsymbol{c}^T \boldsymbol{x}^s - q^* \\ &\geq \frac{1}{2}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*). \end{aligned} \tag{40}$$

Combining (38), (39), and (40), we have

$$\begin{aligned} \mathbb{E}[F(\boldsymbol{x}^{s+1})] - F(\boldsymbol{x}^s) &\leq \frac{1}{K} \min_{a \in [0,1]} -\frac{a}{2}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*) + \frac{2\eta_t M \theta a^2}{2}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*)^2 \\ &= \begin{cases} -1/(16\eta_t M\theta K) & , \ 1/(4\eta_t M\theta(\boldsymbol{c}^T \boldsymbol{x}^s - q^*)) \leq 1 \\ -\frac{1}{4K}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*) & , \ o.w. \end{cases} \end{aligned}$$

Furthermore, we have

$$-\frac{1}{16\eta_t M\theta K} \leq -\frac{1}{16\eta_t M\theta K(F^0 - F^*)}\left(F(\boldsymbol{x}^*) - F^*\right)$$

where $F^0 = F(\boldsymbol{x}^0)$, and

$$-\frac{1}{4K}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*) \leq -\frac{1}{6K}(F(\boldsymbol{x}^s) - F^*)$$

since $F(\boldsymbol{x}^s) - F^* \leq |g(\bar{\Phi}^T \boldsymbol{x}^s) - g(\boldsymbol{p}^*)| + \boldsymbol{c}^T \boldsymbol{x}^s - q^* \leq \frac{3}{2}(\boldsymbol{c}^T \boldsymbol{x}^s - q^*)$. In summary, for Case 1 we obtain

$$\mathbb{E}[F(\boldsymbol{x}^{s+1})] - F^* \leq (1 - \frac{1}{K\gamma_1})\left(\mathbb{E}[F(\boldsymbol{x}^s)] - F^*\right) \tag{41}$$

where

$$\gamma_1 = \max\left\{16\eta_t M\theta(F^0 - F^*)\ ,\ 6\right\}. \tag{42}$$

Case 2: $4L_g^2 \|\bar{\Phi}^T \boldsymbol{x}^s - \boldsymbol{p}^*\|^2 \geq (\boldsymbol{c}^T \boldsymbol{x}^s - q^*)^2$.

In this case, we have

$$\|\boldsymbol{x}^s - \boldsymbol{x}^*\|^2 \leq \theta\left(1 + 4L_g^2\right)\|\bar{\Phi}^T \boldsymbol{x}^s - \boldsymbol{p}^*\|^2, \tag{43}$$

and by strong convexity of $g(\boldsymbol{z})$,

$$F(\boldsymbol{x}^s) - F^* \geq \boldsymbol{c}^T(\boldsymbol{x}^s - \boldsymbol{x}^*) + \nabla g(\boldsymbol{p}^*)^T \bar{\Phi}^T (\boldsymbol{x}^s - \boldsymbol{x}^*) + \frac{\eta_t}{2}\|\bar{\Phi}^T \boldsymbol{x}^s - \boldsymbol{p}^*\|^2.$$

Adding inequality $0 = h(\boldsymbol{x}^s) - h(\boldsymbol{x}^*) \geq \langle \boldsymbol{\rho}^*, \boldsymbol{x}^s - \boldsymbol{x}^* \rangle$ for some $\boldsymbol{\rho}^* \in \partial h(\boldsymbol{x}^*)$ to the above gives

$$F(\boldsymbol{x}^s) - F^* \geq \frac{\eta_t}{2}\|\bar{\Phi}^T \boldsymbol{x}^s - \boldsymbol{p}^*\|^2 \tag{44}$$

since $\boldsymbol{\rho}^* + \boldsymbol{c} + \nabla g(\boldsymbol{p}^*)^T \bar{\Phi}^T = \boldsymbol{\rho}^* + \nabla F(\boldsymbol{x}^*) = 0$. Combining (38), (43), and (44), we obtain

$$\begin{aligned} \mathbb{E}[F(\boldsymbol{x}^{s+1})] - F(\boldsymbol{x}^s) &\leq \frac{1}{K} \min_{a \in [0,1]} -a(F(\boldsymbol{x}^s) - F^*) + \frac{M\theta(1 + 4L_g^2)a^2}{2}\left(F(\boldsymbol{x}^s) - F^*\right) \\ &= -\frac{1}{2M\theta(1 + 4L_g^2)K}\left(F(\boldsymbol{x}^s) - F^*\right) \end{aligned} \tag{45}$$

Combining results of Case 1 (41) and Case 2 (45), and taking expectation on both sides w.r.t. the history leads to the result. □

Theorem 2-2 (Dual-RBCD for Polyhedral Loss)**.** *Let the sequence* $\{\boldsymbol{\alpha}^s\}_{s=1}^{\infty}$ *be the iterates produced by RBCD in the inner loop of Algorithm 2, and* K *be the number of blocks. Denote* $\tilde{F}^*(\boldsymbol{\alpha})$ *as the augmented dual objective function* (18) *and* $\tilde{F}^*_{opt}$ *the optimum of* (18)*. With probability* $1 - \rho$,

$$\tilde{F}^*(\boldsymbol{\alpha}^s) - \tilde{F}^*_{opt} \leq \epsilon, \ \ for \ \ s \geq \gamma K \log(\frac{\tilde{F}^*(\boldsymbol{\alpha}^0) - \tilde{F}^*_{opt}}{\rho\epsilon}) \tag{46}$$

for some constant γ *if* $L_n(.)$ *and* $R(.)$ *are polyhedral.*

Proof. This simply applies Theorem 1 of [13] to transfer the linear convergence in expectation into high-probability iteration complexity. □

6.2 Convergence of Proximal-Point Method

The proof of Theorem 3 comprises two parts. The first part proves linear convergence of Proximal-Point update under assumption that both loss $L_n(.)$ and regularizer $R(.)$ are either strictly convex and smooth or polyhedral. The second part proves a sublinear-type convergence depending on parameter η that holds for general convex function. The second part can be found in, for example, Theorem 2 of [18]. Here we prove the first part.

Here we prove linear convergence of ALM on problem (25) by leveraging some lemmas provided in the recent advance of analysis for Alternating Direction Method of Multiplier (ADMM) [5]. In particular, by taking Proximal-Point updates (or, equivalently, the ALM updates) as performing gradient descent on the convex, smooth function

$$G(\tilde{\boldsymbol{w}}) = \min_{\boldsymbol{w}} \sum_n L_n(\bar{\Phi}_n \boldsymbol{w}) + R(\boldsymbol{w}) + \frac{1}{2\eta}\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|^2 \tag{47}$$

and utilizing error bound proved in [5], we show that the Proximal-Point method linearly converges to the optimum of objective (25).

The following lemma establishes the fact $G(\tilde{\boldsymbol{w}})$ is smooth and its gradient $\nabla G(\tilde{\boldsymbol{w}})$ is Lipschitz continuous with modulus $\frac{1}{\eta}$.

Lemma 4. *The gradient of $G(\tilde{\boldsymbol{w}})$ is of the form*

$$\nabla G(\tilde{\boldsymbol{w}}) = -(\sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n(\tilde{\boldsymbol{w}}) - \boldsymbol{\mu}(\tilde{\boldsymbol{w}})) \tag{48}$$

where $\boldsymbol{\alpha}_n(\tilde{\boldsymbol{w}})$, $\boldsymbol{\mu}(\tilde{\boldsymbol{w}})$ are minimizers of (28)*. Furthermore, the gradient $\nabla G(\tilde{\boldsymbol{w}})$ is Lipschitz continuous with modulus $\frac{1}{\eta}$.*

Proof. The convex objective function (25) fits the form of objective investigated in Multi-block ADMM [5]. Therefore, the theorem follows directly from Lemma 2.1, 2.2 of [5] respectively. □

As a result of Lemma 4, the proximal-point update is exactly gradient descent of step size η, which when performed on a smooth function $G(\tilde{\boldsymbol{w}})$, guarantees descent amount

$$G(\boldsymbol{w}^{t+1}) - G(\boldsymbol{w}^t) \le -\frac{\eta\|\nabla G(\boldsymbol{w}^t)\|^2}{2}. \tag{49}$$

The following theorem then guarantees linear convergence of ALM on our objective (25).

Theorem 6. *Denote S as the set of optimal solutions to* (47) *and $\Pi_S(\boldsymbol{w})$ as the projection of $\boldsymbol{w}$ to S, and let G^* be the optimal function value. The iterates $\{\boldsymbol{w}^t\}_{t=1}^{\infty}$ produced by proximal-point method have*

$$\|\sum_{n=1}^{N} \Phi_n^T \boldsymbol{\alpha}_n(\tilde{\boldsymbol{w}}) - \boldsymbol{\mu}(\tilde{\boldsymbol{w}})\| = \|\nabla G(\boldsymbol{w}^t)\| \le \epsilon$$

for number of iterations

$$t \ge \frac{4\tau}{\eta} \log(\sqrt{\frac{2(G(\boldsymbol{w}^0) - G^*)}{\eta}}\frac{1}{\epsilon}),$$

where $\tau > 0$ is a constant depending on S and initial distance to optimal set $\|\boldsymbol{w}^0 - \Pi_S(\boldsymbol{w}^0)\|$.

Proof. Since $L_n(.)$ and $R(.)$ are either strictly convex and smooth or polyhedral, $L_n^*(.)$ and $R^*(.)$ are also strictly convex and smooth or polyhedral. Therefore, problem (25) satisfies Assumption A(a)-A(e) of [5], and thus the error bound

$$G(\tilde{\boldsymbol{w}}) - G^* \le \tau\|\nabla G(\tilde{\boldsymbol{w}})\|^2 \tag{50}$$

in Lemma 3.1 of [5] applies to $G(\tilde{\boldsymbol{w}})$ with compact domain $\tilde{\boldsymbol{w}} \in R(\boldsymbol{w}^0)$, where $\tau > 0$ is a constant that depends on geometry of S and the initial distance to the set of optimal solutions, and

$$R(\boldsymbol{w}^0) = \left\{\tilde{\boldsymbol{w}} \mid \|\tilde{\boldsymbol{w}} - \Pi_S(\tilde{\boldsymbol{w}})\| \le \|\boldsymbol{w}^0 - \Pi_S(\boldsymbol{w}^0)\|\right\}.$$

is the set of $\tilde{\boldsymbol{w}}$ that lie within a radius of $\|\boldsymbol{w}^0 - \Pi_S(\boldsymbol{w}^0)\|$ to the set S. Note the iterates $\{\boldsymbol{w}^t\}_{t=0}^{\infty}$ all lie in the set $R(\boldsymbol{w}^0)$ by the non-expansiveness of proximal operation. Therefore, the error bound (50) applies to all iterates. Combining (69) and (50), we have

$$G(\boldsymbol{w}^{t+1}) - G(\boldsymbol{w}^t) \leq -\frac{\eta(G(\boldsymbol{w}^t) - G^*)}{2\tau},$$

which implies linear convergence. Let $\Delta G_t = G(\boldsymbol{w}^t) - G^*$, and we have

$$\Delta G_t \leq (1 - \frac{\eta}{2\tau})^t \Delta G_0 \leq e^{-\frac{\eta t}{2\tau}} \Delta G_0 \leq \epsilon_1$$

when

$$t \geq \frac{2\tau}{\eta} \log(\frac{\Delta G_0}{\epsilon_1}).$$

Furthermore, by smoothness of $\nabla G(.)$, we have

$$\Delta G_t \geq \frac{\eta \|\nabla G(\boldsymbol{w}^t)\|^2}{2}.$$

Therefore, to guarantee $\|\nabla G(\boldsymbol{w}^t)\| \leq \epsilon_2$, it suffices to have

$$\Delta G^t \leq \eta \epsilon_2^2 / 2,$$

which can be guaranteed by running

$$t \geq \frac{4\tau}{\eta} \log(\sqrt{\frac{2\Delta G_0}{\eta}} \frac{1}{\epsilon_2})$$

iterations. □

Theorem 7 (Inexact Proximal Map)**.** *Suppose, for a given dual iterate* $\mathbf{w}^t$*, each sub-problem* (11) *is solved inexactly s.t. the solution* $\hat{\boldsymbol{w}}^{t+1}$ *has*

$$\|\hat{\boldsymbol{w}}^{t+1} - \mathbf{prox}_{\eta_t F}(\boldsymbol{w}^t)\| \leq \epsilon_0. \tag{51}$$

Then let $\{\hat{\boldsymbol{w}}^t\}_{t=1}^{\infty}$ *be the sequence of iterates produced by inexact proximal updates and* $\{\boldsymbol{w}^t\}_{t=1}^{\infty}$ *as that generated by exact updates. After t iterations, we have*

$$\|\hat{\boldsymbol{w}}^t - \boldsymbol{w}^t\| \leq t\epsilon_0. \tag{52}$$

Proof. By the non-expansiveness of proximal operation,

$$\begin{aligned}\|\hat{\boldsymbol{w}}^{t+1} - \boldsymbol{w}^{t+1}\| &\leq \|\hat{\boldsymbol{w}}^{t+1} - \mathbf{prox}_{\eta_t F}(\hat{\boldsymbol{w}}^t)\| + \|\mathbf{prox}_{\eta_t F}(\hat{\boldsymbol{w}}^t) - \boldsymbol{w}^{t+1}\| \\ &\leq \epsilon_0 + \|\mathbf{prox}_{\eta_t F}(\hat{\boldsymbol{w}}^t) - \mathbf{prox}_{\eta_t F}(\boldsymbol{w}^t)\| \\ &\leq \epsilon_0 + \|\hat{\boldsymbol{w}}^t - \boldsymbol{w}^t\|.\end{aligned}$$

Recursively applying the above inequality leads to the conclusion. □

7 Appendix-B. ADMM under Limited Memory

In this section, we show how an algorithm for *distributed optimization* can be adapted for *limited-memory* learning, which then serves as a baseline to methods specially designed for limited-memory environment. In particular, the adaption sequentializes parallel computation performed on multiple machines into a series of tasks performed on single machine, where states and data partition of each simulated machine are loaded from (saved to) secondary storage units beforehand (afterward). As an example, we show how to adapt *Alternating Direction Method of Multiplier (ADMM)*, a recently proposed distributed optimization framework [1], into our setting.

Given a problem of the form

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{min} \quad \sum_{i=1}^{N} f_i(\boldsymbol{w}), \tag{53}$$

Algorithm 3 ADMM (limited memory)

```
1. Split data $\mathcal{D}$ into blocks $B_1, B_2, ..., B_K$.
2. Initialize $\boldsymbol{w}_k^0 = \mathbf{0}$, $\boldsymbol{z}^0 = \mathbf{0}$, $\boldsymbol{\mu}_k^0 = \mathbf{0}$.
for $t = 0, 1, ...$ (outer iteration) do
    3. $\boldsymbol{z}^{t+1} = \mathbf{0}$
    for $k = 1, 2, ..., K$ do
        4.1. Swap data block $B_k$, $\boldsymbol{w}_k$, $\boldsymbol{\mu}_k$ into memory.
        4.2. $\boldsymbol{w}_k^{t+1} = argmin_{\boldsymbol{w}}\ \mathcal{L}_k(\boldsymbol{w}, \boldsymbol{z}^t, \boldsymbol{\mu}_k^t)$
        4.3. $\boldsymbol{z}^{t+1} += (\boldsymbol{w}_k^{t+1} + \boldsymbol{\mu}_k^t/\rho)/K$
    end for
    for $k = 1, 2, ..., K$ do
        5.1. Swap $\boldsymbol{w}_k^{t+1}$, $\boldsymbol{\mu}_k^t$ into memory.
        5.2. $\boldsymbol{\mu}_k^{t+1} = \boldsymbol{\mu}_k^t + \eta(\boldsymbol{w}_k^{t+1} - \boldsymbol{z}^{t+1})$.
    end for
end for
```

Algorithm 4 Block-Coordinate ADMM (BC-ADMM)

```
1. Split data $\mathcal{D}$ into blocks $B_1, B_2, ..., B_K$.
2. Initialize $\boldsymbol{w}_k^0 = \mathbf{0}$, $\boldsymbol{z}^0 = \mathbf{0}$, $\boldsymbol{\mu}_k^0 = \mathbf{0}$.
for $t = 0, 1, ...$ do
    3.1. Randomly chosen $k \in \{1..K\}$ w/o replacement.
    3.2. Swap data block $B_k$, $\boldsymbol{w}_k^t$, $\boldsymbol{\mu}_k^t$ into memory.
    3.3. $\boldsymbol{\mu}_k^{t+1} = \boldsymbol{\mu}_k^t + \eta(\boldsymbol{w}_k^t - \boldsymbol{z}^t)$.
    3.4. $\boldsymbol{w}_k^{t+1} = argmin_{\boldsymbol{w}}\ \mathcal{L}_k(\boldsymbol{w}, \boldsymbol{z}^t, \boldsymbol{\mu}_k^{t+1})$
    3.5. $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + (\boldsymbol{w}_k^{t+1} + \boldsymbol{\mu}_k^{t+1}/\rho)/K - (\boldsymbol{w}_k^t + \boldsymbol{\mu}_k^t/\rho)/K$
end for
```

the ADMM framework splits (53) into K smaller sub-problems defined on different data blocks $B_1, B_2, ..., B_K$, and formulate the dual problem of

$$\begin{aligned} \underset{\boldsymbol{w}_k, \boldsymbol{z}}{min} \quad & \sum_{k=1}^{K} f_k(\boldsymbol{w}_k) + \frac{\rho}{2}(\boldsymbol{w}_k - \boldsymbol{z})^2 \\ s.t. \quad & \boldsymbol{w}_k - \boldsymbol{z} = 0,\ k = 1, .., K\ , \end{aligned} \tag{54}$$

where $f_k(\boldsymbol{w}_k) = \sum_{i \in B_k} f_i(\boldsymbol{w}_k)$, $\boldsymbol{z}$ is the *consensus parameters*, and $\rho > 0$ is a hyper-parameter. The ADMM procedure finds the saddle point of Lagrangian

$$\underset{\boldsymbol{\mu}_k}{max} \quad \underset{\boldsymbol{w}_k, \boldsymbol{z}}{min} \quad \mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\mu}) = \sum_{k=1}^{K} f(\boldsymbol{w}_k) + \boldsymbol{\mu}_k^T(\boldsymbol{w}_k - \boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{w}_k - \boldsymbol{z}\|^2 \tag{55}$$

via the following iterate

$$\boldsymbol{w}^{t+1} = \underset{\boldsymbol{w}}{argmin}\ \mathcal{L}(\boldsymbol{w}, \boldsymbol{z}^t, \boldsymbol{\mu}^t) \tag{56}$$

$$\boldsymbol{z}^{t+1} = \underset{\boldsymbol{z}}{argmin}\ \mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}, \boldsymbol{\mu}^t) \tag{57}$$

$$\boldsymbol{\mu}_k^{t+1} = \boldsymbol{\mu}_k^t + \eta(\boldsymbol{w}_k^{t+1} - \boldsymbol{z}^{t+1}),\ k = 1, ..., K, \tag{58}$$

where η is a constant step size. Since given $\boldsymbol{z}^t$, $\mathcal{L}(\boldsymbol{w}, \boldsymbol{z}^t, \boldsymbol{\mu}^t)$ is separable w.r.t. $\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_K$, step (56) can be solved separately for each $\boldsymbol{w}_k$ as

$$\boldsymbol{w}_k^{t+1} = \underset{\boldsymbol{w}_k}{argmin}\ \mathcal{L}_k(\boldsymbol{w}_k, \boldsymbol{z}^t, \boldsymbol{\mu}_k^t),\ k = 1, .., K. \tag{59}$$

Since the bottleneck of iterate lies in (59), ADMM is inherently suitable for distributed optimization via solving the K subproblems (59) on K machines. The only step requiring communication is (57), which has close-form solution

$$\boldsymbol{z}^{t+1} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{w}_k^{t+1} + \boldsymbol{\mu}_k^t/\rho, \tag{60}$$

that is, a simple average over parameters and multipliers. In limited-memory environment, however, only one block B_k of samples can be fit into memory at a time, and thus K times of swapping is required at each iteration. A naive implementation is depicted in Algorithm 3. Note, in some high-dimensional problem, the model parameters $\boldsymbol{w}_k$ and $\boldsymbol{\mu}_k$ can be of comparable size to the data block, and thus need to be stored out of memory. One drawback of algorithm 3 is that the *consensus parameter* $\boldsymbol{z}$ is not updated until K subproblems are solved. In Algorithm 4, we propose another adaption that updates the dual variables of one randomly chosen block B_k at a time as follows

$$\boldsymbol{\mu}_k^t = \boldsymbol{\mu}_k^{t-1} + \eta(\boldsymbol{w}_k^t - \boldsymbol{z}^t) \tag{61}$$

$$\boldsymbol{w}_k^{t+1} = \underset{\boldsymbol{w}_k}{argmin}\ \mathcal{L}_k(\boldsymbol{w}_k, \boldsymbol{z}^t, \boldsymbol{\mu}_k^t) \tag{62}$$

$$\boldsymbol{z}^{t+1} = \underset{\boldsymbol{z}}{argmin}\ \mathcal{L}(\boldsymbol{w}_1^t, ..., \boldsymbol{w}_k^{t+1}, ..., \boldsymbol{w}_K^t, \boldsymbol{z}, \boldsymbol{\mu}_k^t). \tag{63}$$

In this version of limited-memory ADMM, the information learnt from one block can be passed to the next subproblem immediately, and consensus parameters $\boldsymbol{\mu}_k$, $\boldsymbol{w}_k$ only need to be swapped once for each iteration. It has been shown that standard ADMM iterates in Algorithm 3 have global linear convergence to the optimum [5]. The following theorem shows the same type of convergence guarantee also applies to Algorithm 4 .

Theorem 8 (BC-ADMM Convergence)**.** *Consider a regularized ERM problem* (53) *of the form*

$$f_i(\boldsymbol{w}) = L_i(\Phi_i\boldsymbol{w}) + \frac{1}{N}R(\boldsymbol{w}).$$

Let $d(\boldsymbol{\mu})$ be the dual function value of problem (54)*. If the loss function $L_i(.)$ is smooth, and one of $L_i(.)$ or $R(.)$ is strongly convex, Algorithm 4 converges to the optimum of* (53) *at a linear rate, that is,*

$$E[\Delta_p^t + \Delta_d^t] \leq \frac{1}{1+\lambda}(\Delta_p^{t-1} + \Delta_d^{t-1}) \tag{64}$$

for some constant $\lambda > 0$, where

$$\begin{aligned} \Delta_p^t &= \mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - d(\boldsymbol{\mu}^t) \\ \Delta_d^t &= d^* - d(\boldsymbol{\mu}^t) \end{aligned} \tag{65}$$

are the primal and dual residuals at iterate t respectively.

Though being effective, the adapted algorithm takes little advantage of the sequential nature of limiter-memory setting. In particular, since the distributed learning algorithm is designed to allow parallel updates, the information passed among parallel sub-problems is limited and the updates on dual variables (58), (62) are conservative with step size η compared to the exact block-coordinate minimization (12) in the Dual-Augmented Block Minimization framework. Note ADMM can be seen as an approximate Gradient Descent method on the dual, while analysis in coordinate descent literature [13] shows that Block-Coordinate descent can be up to K times faster than Gradient Descent in the worst-conditioned case, where K is the number of blocks.

8 Appendix-C. Convergence of Block-Coordinate ADMM

Let $d(\boldsymbol{\mu}) = \min_{\boldsymbol{w},\boldsymbol{z}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\mu})$ be the dual objective for $\boldsymbol{\mu}$ and $d^* = \max_{\boldsymbol{\mu}} d(\boldsymbol{\mu})$ be the optimal dual objective value, we define primal residual Δ_p^t and dual residual Δ_d^t of current iterate $(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t)$ as

$$\begin{aligned} \Delta_p^t &= \mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - d(\boldsymbol{\mu}^t) \\ \Delta_d^t &= d^* - d(\boldsymbol{\mu}^t). \end{aligned} \tag{66}$$

Note $\Delta_p^t \geq 0$, $\Delta_d^t \geq 0$, and $\Delta_d^t = \Delta_p^t = 0$ if only if $(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t)$ is optimal.

Lemma 5 (Dual Iterate)**.** *For all* $t \geq 1$*,*

$$\Delta_d^t - \Delta_d^{t-1} \leq -\eta(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t),$$

where $(\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{z}}^t)$ *is the solution to* $\min_{\boldsymbol{w},\boldsymbol{z}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\mu}^t)$ *that is closest to* $(\boldsymbol{w}^t, \boldsymbol{z}^t)$*.*

Proof.

$$\begin{aligned}\Delta_d^t - \Delta_d^{t-1} &= d(\boldsymbol{\mu}^{t-1}) - d(\boldsymbol{\mu}^t)\\ &= \mathcal{L}(\bar{\boldsymbol{w}}^{t-1}, \bar{\boldsymbol{z}}^{t-1}, \boldsymbol{\mu}^{t-1}) - \mathcal{L}(\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{z}}^t, \boldsymbol{\mu}^t)\\ &\leq \mathcal{L}(\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{z}}^t, \boldsymbol{\mu}^{t-1}) - \mathcal{L}(\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{z}}^t, \boldsymbol{\mu}^t)\\ &= (\boldsymbol{\mu}^{t-1} - \boldsymbol{\mu}^t)^T(\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{z}}^t)\\ &= (\boldsymbol{\mu}_k^{t-1} - \boldsymbol{\mu}_k^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\\ &= -\eta(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t),\end{aligned}$$

where the third inequality follows from definition $(\bar{\boldsymbol{w}}^{t-1}, \bar{\boldsymbol{z}}^{t-1}) = argmin_{\boldsymbol{w},\boldsymbol{z}}\mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\mu}^{t-1})$. □

Lemma 6 (Primal Iterate)**.** *For all* $t \geq 1$*,*

$$\begin{aligned}\Delta_p^t - \Delta_p^{t-1} \leq& -\rho\left(\|\boldsymbol{w}_k^{t+1} - \boldsymbol{w}_k^t\|^2 + \|\boldsymbol{z}^{t+1} - \boldsymbol{z}^t\|^2\right)\\ &+ \eta\left(\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2 - (\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\right)\end{aligned}$$

Proof.

$$\begin{aligned}&\Delta_p^t - \Delta_p^{t-1} =\\ &\left(\mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - d(\boldsymbol{\mu}^t)\right) - \left(\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^{t-1}) - d(\boldsymbol{\mu}^{t-1})\right),\end{aligned}$$

where $d(\boldsymbol{\mu}^{t-1}) - d(\boldsymbol{\mu}^t)$ can be obtained via Lemma 2.1 as

$$d(\boldsymbol{\mu}^{t-1}) - d(\boldsymbol{\mu}^t) = -\eta(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t). \tag{67}$$

It remains to find

$$\begin{aligned}&\mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^{t-1}) =\\ &\mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t) + \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^{r-1}).\end{aligned}$$

From strong convexity of the augmented term $\frac{\rho}{2}\|\boldsymbol{w}_k - \boldsymbol{z}\|^2$, and that $\boldsymbol{w}^{t+1}$, $\boldsymbol{z}^{t+1}$ are minimizers for (56) and (57) respectively, we can bound the primal descent amount by

$$\begin{aligned}&\mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t)\\ &= \mathcal{L}_k(\boldsymbol{w}_k^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}_k^t) - \mathcal{L}_k(\boldsymbol{w}_k^t, \boldsymbol{z}^t, \boldsymbol{\mu}_k^t)\\ &\leq -\rho\left(\|\boldsymbol{w}_k^{t+1} - \boldsymbol{w}_k^t\|^2 + \|\boldsymbol{z}^{t+1} - \boldsymbol{z}^t\|^2\right).\end{aligned}$$

It is also known that

$$\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^{r-1}) = \eta\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2.$$

Therefore,

$$\begin{aligned}&\mathcal{L}(\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}, \boldsymbol{\mu}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\mu}^{t-1})\\ &\leq -\rho\left(\|\boldsymbol{w}_k^{t+1} - \boldsymbol{w}_k^t\|^2 + \|\boldsymbol{z}^{t+1} - \boldsymbol{z}^t\|^2\right) + \eta\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2.\end{aligned}$$

Combine above inequality with (67), we obtain the conclusion. □

The following theorem guarantees descent of the primal-dual residual $\Delta_p^t + \Delta_d^t$ in expectation for each iteration of BC-ADMM.

Theorem 9 (Guaranteed Descent)**.** *For step-size* η *sufficiently small,*

$$E[\Delta_p^t + \Delta_d^t] < (\Delta_p^{t-1} + \Delta_d^{t-1})$$

for all $t \geq 1$*, where* $E[.]$ *is expectation over blocks* k_1*,* k_2*, ...,* k_R *drawn at iteration* t*.*

Proof. Define

$$\Delta \boldsymbol{z}_k^t = (\boldsymbol{w}_k^{t+1} + \boldsymbol{\mu}_k^t/\rho)/K - (\boldsymbol{w}_k^t + \boldsymbol{\mu}_k^{t-1}/\rho)/K$$

and

$$\Delta \boldsymbol{z}^t = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{w}_k^{t+1} + \boldsymbol{\mu}_k^t/\rho) - \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{w}_k^t + \boldsymbol{\mu}_k^{t-1}/\rho)$$

By Lemma 2.1 and 2.2, we have

$$\begin{aligned}&(\Delta_p^t + \Delta_d^t) - (\Delta_p^{t-1} + \Delta_d^{t-1})\\ &= (\Delta_p^t - \Delta_p^{t-1}) + (\Delta_d^t - \Delta_d^{t-1})\\ &\le -\rho\left(\|\Delta \boldsymbol{w}_k^t\|^2 + \|\Delta \boldsymbol{z}_k^t\|^2\right)\\ &\quad + \eta\left(\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2 - 2(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\right).\end{aligned}$$

Taking expectation on both sides w.r.t. the random selected indexes k_1, k_2, ..., k_R, we have

$$\begin{aligned}&E[\Delta_p^t + \Delta_d^t] - (\Delta_p^{t-1} + \Delta_d^{t-1})\\ &\le -\frac{\rho R}{K}\left(\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2\right)\\ &+ \frac{\eta R}{K}\left(\sum_{k=1}^{K}\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2 - 2\sum_{k=1}^{K}(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\right),\end{aligned}$$

where $\Delta \boldsymbol{w}^t$ and $\Delta \boldsymbol{z}^t$ are the primal iterate of standard ADMM, and

$$\begin{aligned}&\sum_{k=1}^{K}\|\boldsymbol{w}_k^t - \boldsymbol{z}^t\|^2 - 2(\boldsymbol{w}_k^t - \boldsymbol{z}^t)^T(\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\\ &= \sum_{k=1}^{K}\|(\boldsymbol{w}_k^t - \boldsymbol{z}^t) - (\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t)\|^2 - \|\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t\|^2\\ &\le 2\sum_{k=1}^{K}\left(\|\boldsymbol{w}_k^t - \bar{\boldsymbol{w}}_k^t\|^2 + \|\boldsymbol{z}^t - \bar{\boldsymbol{z}}^t\|^2\right) - \|\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t\|^2.\end{aligned}$$

Now we invoke the error bound in [5, Lemma 2.3, 2.5] to bound the distance between $(\boldsymbol{w}^t, \boldsymbol{z}^t)$ and $(\bar{\boldsymbol{w}}^t, \bar{\boldsymbol{z}}^t)$ in terms of progress in primal iterate $\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2$ as

$$\sum_{k=1}^{K}\|\boldsymbol{w}^t - \bar{\boldsymbol{w}}^t\|^2 + \|\boldsymbol{z}^t - \bar{\boldsymbol{z}}^t\|^2 \le \tau(\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2), \tag{68}$$

where τ is a positive constant. Then we have

$$\begin{aligned}&E[\Delta_p^t + \Delta_d^t] - (\Delta_p^{t-1} + \Delta_d^{t-1})\\ &\le -\frac{R(\rho - 2\eta\tau)}{K}(\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2) - \frac{R\eta}{K}\sum_{k=1}^{K}\|\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t\|^2,\end{aligned} \tag{69}$$

which is always negative for $\eta < \rho/(2\tau)$. □

Then we can have following theorem for linear convergence of BC-ADMM.

Theorem 10 (BC-ADMM Convergence)**.** *Consider a regularized ERM problem* (53) *of the form*

$$f_i(\boldsymbol{w}) = L_i(\Phi_i \boldsymbol{w}) + \frac{1}{N}R(\boldsymbol{w}).$$

If the loss function $L_i(.)$ is smooth, and one of $L_i(.)$ or $R(.)$ is strongly convex, Algorithm 4 converges to the optimum of (53) *at a linear rate, that is,*

$$E[\Delta_p^t + \Delta_d^t] \le \frac{1}{1+\lambda}(\Delta_p^{t-1} + \Delta_d^{t-1}) \tag{70}$$

for some constant $\lambda > 0$.

Proof. To prove linear convergence, we show that the two terms in (69) can be lower bounded by the current residual Δ_p^t, Δ_d^t respectively. In particular, we invoke the error bound in [5, Lemma 3.1] that shows

$$\Delta_d^t \leq \tau_2 \|\nabla d(\boldsymbol{\mu}^t)\| = \tau_2 \|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{z}}^t\|^2 \tag{71}$$

and

$$\Delta_p^t \leq \xi \left(\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2\right) \tag{72}$$

for some positive constant τ_2, ξ and $\forall t \geq t_0$, where (72) has combined [5, Lemma 3.1] and (68). Apply above error bounds on (69), we have

$$\begin{aligned}
&E[\Delta_p^t + \Delta_d^t] - (\Delta_p^{t-1} + \Delta_d^{t-1}) \\
&\leq -\frac{R(\rho - 2\eta\tau)}{K}(\|\Delta \boldsymbol{w}^t\|^2 + \|\Delta \boldsymbol{z}^t\|^2) - \frac{R\eta}{K}\sum_{k=1}^{K}\|\bar{\boldsymbol{w}}_k^t - \bar{\boldsymbol{z}}^t\|^2 \\
&\leq -\frac{R(\rho - 2\eta\tau)}{K\xi}\Delta_p^t - \frac{R\eta}{K\tau_2}\Delta_d^t \\
&\leq -\lambda(\Delta_p^t + \Delta_d^t),
\end{aligned}$$

where $\lambda = \frac{R}{K}\min\left\{(\rho - 2\eta\tau)\xi^{-1}, \eta\tau_2^{-1}\right\} > 0$ for step size $\eta < \rho/(2\tau)$. After rearrangement we have

$$E[\Delta_p^t + \Delta_d^t] \leq \frac{1}{1+\lambda}(\Delta_p^{t-1} + \Delta_d^{t-1}), \quad t \geq t_0.$$

□