

ISE



Industrial and
Systems Engineering

Adding vs. Averaging in Distributed Primal-Dual Optimization

CHENXIN MA¹, VIRGINIA SMITH², MARTIN JAGGI³, MICHAEL I.
JORDAN², PETER RICHTÁRIK⁴, AND MARTIN TAKÁČ¹

¹Lehigh University ²UC Berkeley ³ETH Zürich ⁴The University of Edinburgh

ISE Technical Report 15T-004



Adding vs. Averaging in Distributed Primal-Dual Optimization

Chenxin Ma

Industrial and Systems Engineering, Lehigh University, USA

CHM514@LEHIGH.EDU

Virginia Smith

University of California, Berkeley, USA

VSMITH@BERKELEY.EDU

Martin Jaggi

ETH Zürich, Switzerland

JAGGI@INF.ETHZ.CH

Michael I. Jordan

University of California, Berkeley, USA

JORDAN@CS.BERKELEY.EDU

Peter Richtárik

School of Mathematics, University of Edinburgh, UK

PETER.RICHTARIK@ED.AC.UK

Martin Takáč

Industrial and Systems Engineering, Lehigh University, USA

TAKAC.MT@GMAIL.COM

Abstract

Distributed optimization algorithms for large-scale machine learning suffer from a communication bottleneck. Reducing communication makes the efficient aggregation of partial work from different machines more challenging. In this paper we present a novel generalization of the recent communication efficient primal-dual coordinate ascent framework (CoCoA). Our framework, CoCoA⁺, allows for *additive* combination of local updates to the global parameters at each iteration, whereas previous schemes only allowed conservative averaging. We give stronger (primal-dual) convergence rate guarantees for both CoCoA as well as our new variants, and generalize the theory for both methods to also cover non-smooth convex loss functions. We provide an extensive experimental comparison on several real-world distributed datasets, showing markedly improved performance, especially when scaling up the number of machines.

1. Introduction

With the wide availability of large datasets that exceed the storage capacity of single machines, distributed optimization methods for machine learning have become in-

creasingly important. Existing methods require significant communication between workers, frequently equaling the amount of local computation (or reading of local data). As a result, distributed machine learning suffers significantly from a communication bottleneck on real world systems, where communication is typically several orders of magnitudes slower than reading data from main memory.

In this work we focus on optimization problems with empirical loss minimization structure, meaning that the objective is a sum of the loss functions of each individual datapoint. This includes the most commonly used regularized variants of linear regression and classification methods. For this class of problems, the recently proposed CoCoA approach (Yang, 2013; Jaggi et al., 2014; Konečný et al., 2014) has formulated a communication efficient primal-dual coordinate ascent scheme which targets the communication bottleneck, by allowing more computation to be performed on data local subproblems on each machine before a single vector is communicated. By appropriately choosing the amount of local computation per round, this framework allows one to control the trade-off between *communication* and *local computation* based on the systems hardware at hand.

However, the performance of CoCoA (as well as the related primal SGD based methods) is slowed down significantly by the need to average updates between all machines. As the number of machines K grows, the updates get diluted and slowed down by $1/K$, e.g. in the case

where all machines except one would already have reached the solutions of their respective partial optimization tasks. A somewhat opposite problem appears if the updates are added instead, as the algorithms can then diverge, as we will observe in the practical experiments below.

To address both described issues, in this paper we develop a novel generalization of the local CoCoA subproblems assigned to each worker, making the framework more powerful in the following sense: Without extra computational cost, the set of locally computed updates from the modified subproblems (one from each machine) can be combined much more efficiently between the machines. The CoCoA⁺ updates proposed here can be aggressively *added* (hence the ‘+’-suffix), which gives a much faster convergence both in practice and also in theory. We will see that the difference is particularly significant as the number of machines K becomes large.

1.1. Contributions

Strong Scaling. To our knowledge, our distributed optimization framework is the first to exhibit favorable *strong scaling* for the class of problems considered when the number of machines K increases, while the data size is kept fixed. More precisely, while the convergence rate of CoCoA degrades as the number of workers K is increased, the stronger theoretical convergence rate here is – in the worst case – *independent* of K . Our practical experiments in Section 7 confirm the improved speed of convergence. Since the number of communicated vectors is only one per round and worker, this favorable scaling might be surprising. Indeed, for existing methods, splitting data among more machines generally increases the requirements for communication (Shamir & Srebro, 2014), which often has severe effects on overall runtime.

Theoretical Analysis for Non-Smooth Functions. While the existing analysis for CoCoA in (Jaggi et al., 2014) only covered smooth loss functions, here we extend the class of functions where the rates apply (to include e.g. Support Vector Machines and non-smooth regression variants). We provide a primal-dual convergence rate for both CoCoA as well as our new method CoCoA⁺ in the case of general convex (L -Lipschitz) losses.

Primal-Dual Convergence Rate. Furthermore, we additionally strengthen the rates by showing stronger primal-dual convergence for both algorithmic frameworks, which are almost tight to their objective-only counterparts. Primal-dual rates for CoCoA had not previously been analyzed in the general convex case. Our primal-dual rates allow efficient and practical certificates for the optimization quality, e.g. for stopping criterions. The new rates apply for both smooth and non-smooth, and for both CoCoA as

well as the extended CoCoA⁺.

Experimental Results. We provide a thorough experimental comparison with competing algorithms, which we all implement in Spark. Our practical results confirm the strong scaling of CoCoA⁺ as the number of machines K grows up to 100, while competing methods – including original CoCoA – are slowing down significantly with larger K .

1.2. History and Related Work

While optimal algorithms for the serial (single machine) case are already very well researched and understood, the literature in the distributed setting is relatively sparse. In particular, the details on optimal trade-offs between computation and communication, as well as optimization or statistical accuracy, are still widely unclear. For an overview over the currently active research field, we refer the reader to (Balcan et al., 2012; Richtárik & Takáč, 2013b; Duchi et al., 2013; Yang, 2013; ?; Liu & Wright, 2014; Fercq & Richtárik, 2013; Fercq et al., 2014; Jaggi et al., 2014; Shamir & Srebro, 2014; Shamir et al., 2014; Zhang & Xiao, 2015; Qu & Richtárik, 2014) and the references therein. We provide a detailed comparison of our proposed framework to the related work in Section 6 below.

2. Setup

We consider regularized empirical loss minimization problems of the following well-established form:

$$\min_{w \in \mathbb{R}^d} \left\{ P(w) := \frac{1}{n} \sum_{i=1}^n \ell_i(x_i^T w) + \frac{\lambda}{2} \|w\|^2 \right\}. \quad (1)$$

Here the vectors $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$ represent the training data examples, and the $\ell_i(\cdot)$ are arbitrary convex real-valued loss functions (e.g. hinge loss), possibly depending on label information for the i -th datapoints. The constant $\lambda > 0$ is the regularization parameter.

The above class of problems includes many standard problems of wide interest in machine learning, statistics and signal processing, including Support Vector Machines, as well as regularized linear and logistic regression, ordinal regression, and others.

Dual Problem, and Primal-Dual Certificates. The conjugate dual optimization problem of (1) takes following form:

$$\max_{\alpha \in \mathbb{R}^n} \left\{ \mathcal{D}(\alpha) := -\frac{1}{n} \sum_{j=1}^n \ell_j^*(-\alpha_j) - \frac{\lambda}{2} \left\| \frac{1}{\lambda n} A \alpha \right\|^2 \right\}, \quad (2)$$

Here the data matrix $A = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ collects all data-points as its columns, and ℓ_j^* is the conjugate

function to ℓ_j . See e.g. (Shalev-Shwartz & Zhang, 2013c) for several concrete applications.

It is possible assign for any dual vector $\alpha \in \mathbb{R}^n$ a corresponding primal feasible point

$$w(\alpha) = \frac{1}{\lambda n} A\alpha. \quad (3)$$

The duality gap function is given as follows

$$G(\alpha) := \mathcal{P}(w(\alpha)) - \mathcal{D}(\alpha). \quad (4)$$

By weak duality, every value $\mathcal{D}(\alpha)$ at a dual candidate α provides a lower bound for every primal value $\mathcal{P}(w)$, therefore the duality gap being a certificate for the approximation quality: The unknown true optimum $\mathcal{P}(w^*)$ must always lie within the duality gap, i.e. $G(\alpha) = \mathcal{P}(w) - \mathcal{D}(\alpha) \geq \mathcal{P}(w) - \mathcal{P}(w^*) \geq 0$.

In large-scale machine learning such as in the setting here, the availability of such a computable measure of approximation quality is a crucial benefit for training methods. In contrast, practitioners using classical primal-only methods such as SGD have no means by which to accurately detect if a model has been well trained, as $\mathcal{P}(w^*)$ is unknown.

Classes of Loss-Functions. In order to simplify the presentation, we assume that all loss functions ℓ_i are always non-negative, and

$$\ell_i(0) \leq 1 \quad \forall i. \quad (5)$$

Definition 1 (L -Lipschitz continuous loss). A function $\ell_i : \mathbb{R} \rightarrow \mathbb{R}$ is L -Lipschitz continuous if $\forall a, b \in \mathbb{R}$, we have

$$|\ell_i(a) - \ell_i(b)| \leq L|a - b|. \quad (6)$$

Definition 2 ($(1/\mu)$ -smooth loss). A function $\ell_i : \mathbb{R} \rightarrow \mathbb{R}$ is $(1/\mu)$ -smooth if it is differentiable and its derivative is $(1/\mu)$ -Lipschitz continuous, i.e. $\forall a, b \in \mathbb{R}$, we have

$$|\ell'_i(a) - \ell'_i(b)| \leq \frac{1}{\mu}|a - b|. \quad (7)$$

3. The CoCoA⁺ Algorithm Framework

Data Partitioning. We write $\{\mathcal{P}_k\}_{k=1}^K$ for the given partition of the datapoints $[n] := \{1, 2, \dots, n\}$ over the K worker machines. We denote the size of each part by $n_k = |\mathcal{P}_k|$. For any $k \in [K]$ and $\alpha \in \mathbb{R}^n$ we use the notation $\alpha_{[k]} \in \mathbb{R}^n$ for the vector

$$(\alpha_{[k]})_i := \begin{cases} 0, & \text{if } i \notin \mathcal{P}_k, \\ \alpha_i, & \text{otherwise.} \end{cases}$$

Local Subproblems in CoCoA⁺. We can define a data-local subproblem of the original dual optimization problem (2), which can be solved on machine k and only requires

accessing data which is already available locally on the machine, i.e. datapoints with $i \in \mathcal{P}_k$.

More formally, we assign to each machine k the following subproblem, only depending on the change in local dual variables α_i with $i \in \mathcal{P}_k$:

$$\max_{\Delta\alpha_{[k]} \in \mathbb{R}^n} \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w) \quad (8)$$

where

$$\begin{aligned} \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w) &:= -\frac{1}{n} \sum_{i \in \mathcal{P}_k} \ell_i^*(-\alpha_i - (\Delta\alpha_{[k]})_i) \\ &\quad - \frac{1}{K} \frac{\lambda}{2} \|w\|^2 - \frac{1}{n} w^T A \Delta\alpha_{[k]} \\ &\quad - \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A \Delta\alpha_{[k]} \right\|^2. \end{aligned} \quad (9)$$

The role of the parameter σ' is to measure the difficulty of the given data partition. For our purposes, we will see that it must be chosen not smaller than

$$\sigma' \geq \sigma'_{min} := \gamma \max_{\alpha \in \mathbb{R}^n} \frac{\|A\alpha\|^2}{\sum_{k=1}^K \|A\alpha_{[k]}\|^2}. \quad (10)$$

We will see (Lemma 3) that this parameter can be upper-bounded by γK , which is trivial to calculate for all values of $\gamma \in \mathbb{R}$. Later on, our main theorems below will show convergence rates for when γ is used as the aggregation parameter in our algorithm framework. Proofs of all statements are provided in the supplementary material.

We show experimentally (Section 7) that the following safe upper bound for σ' has a minimal effect on overall performance of the algorithm.

Lemma 3. *The choice of $\sigma' := \gamma K$ is valid for (10), i.e.*

$$\gamma K \geq \sigma'_{min}.$$

Interpretation. The above definition of the local subproblems with objective functions $\mathcal{G}_k^{\sigma'}$ are such that they closely approximate the dual objective \mathcal{D} , as we vary the ‘local’ variable $\Delta\alpha_{[k]}$, in the following precise sense:

Lemma 4. *For any $\alpha, \Delta\alpha_{[1]}, \dots, \Delta\alpha_{[K]} \in \mathbb{R}^n$ it holds that*

$$\mathcal{D}\left(\alpha + \gamma \sum_{k=1}^K \Delta\alpha_{[k]}\right) \geq (1-\gamma)\mathcal{D}(\alpha) + \gamma \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w). \quad (11)$$

Notion of Approximation Quality of the Local Solver.

Assumption 1 (Θ -approximate solution). *We assume that there exists $\Theta \in [0, 1)$ such that $\forall k \in [K]$, the local solver*

at any iteration t produces a (possibly) randomized approximate solution $\Delta\alpha_{[k]}$, which satisfies

$$\mathbb{E}[\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w)] \leq \quad (12)$$

$$\Theta\left(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w)\right),$$

where

$$\Delta\alpha^* \in \arg \min_{\Delta\alpha \in \mathbb{R}^n} \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w). \quad (13)$$

We are now ready to describe the CoCoA⁺ algorithm framework, shown in Algorithm 1. The crucial difference compared to the existing CoCoA algorithm (Jaggi et al., 2014) is the more general subproblem, as defined in (9), as well as the aggregation parameter γ . These modifications allow for updates to the global vector w to be added directly.

Algorithm 1 CoCoA⁺ Framework

- 1: **Input:** Datapoints A distributed according to partition $\{\mathcal{P}_k\}_{k=1}^K$. Aggregation parameter $\gamma \in [\frac{1}{K}, 1]$, subproblem parameter σ' for the subproblems $\mathcal{G}^{\sigma'}(\Delta\alpha_{[k]}, w)$ for each $k \in [K]$. Starting point $\alpha^{(0)} = \mathbf{0} \in \mathbb{R}^n$, $w^{(0)} := \mathbf{0} \in \mathbb{R}^d$.
 - 2: **for** $t = 0, 1, 2, \dots$ **do**
 - 3: **for** $k \in \{1, 2, \dots, K\}$ **in parallel over computers do**
 - 4: call the local solver, computing a Θ -approximate solution $\Delta\alpha_{[k]}$ of the subproblem (9)
 - 5: update $\alpha_{[k]}^{(t+1)} := \alpha_{[k]}^{(t)} + \gamma \Delta\alpha_{[k]}$
 - 6: return $\Delta w_k := \frac{1}{\lambda n} A \Delta\alpha_{[k]}$
 - 7: **end for**
 - 8: reduce $w^{(t+1)} := w^{(t)} + \gamma \sum_{k=1}^K \Delta w_k$. (14)
 - 9: **end for**
-

4. Convergence Guarantees

Before being able to state our main convergence results, we introduce some useful quantities and the following main lemma characterizing the effect of iterations of Algorithm 1, for any chosen internal local solver.

Lemma 5. *Let ℓ_i^* be strongly¹ convex with convexity parameter $\mu \geq 0$ with respect to the norm $\|\cdot\|$, $\forall i \in [n]$. Then for all iterations t of Algorithm 1 under Assumption 1, and any $s \in [0, 1]$, it holds that*

$$\mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^{(t)})] \geq \quad (15)$$

$$\gamma(1 - \Theta) \left(sG(\alpha^{(t)}) - \frac{\sigma'}{2\lambda} \left(\frac{s}{n}\right)^2 R^{(t)} \right),$$

¹Note that the case of weakly convex $\ell_i^*(\cdot)$ is explicitly allowed here as well, as the Lemma holds for the case $\mu = 0$.

where

$$R^{(t)} := -\frac{\lambda\mu n(1-s)}{\sigma' s} \|u^{(t)} - \alpha^{(t)}\|^2 \quad (16)$$

$$+ \sum_{k=1}^K \|A(u^{(t)} - \alpha^{(t)})_{[k]}\|^2,$$

$u^{(t)} = (u_1^{(t)}, \dots, u_n^{(t)})^T$ and

$$-u_i^{(t)} \in \partial\ell_i(w(\alpha^{(t)})^T x_i). \quad (17)$$

The following lemma provides a uniform bound on $R^{(t)}$:

Lemma 6. *Assuming the ℓ_i are L -Lipschitz continuous for all $i \in [n]$, then*

$$\forall t : R^{(t)} \leq 4L^2 \underbrace{\sum_{k=1}^K \sigma_k n_k}_{=: \sigma}, \quad (18)$$

where

$$\sigma_k := \max_{\alpha_{[k]} \in \mathbb{R}^n} \frac{\|A\alpha_{[k]}\|^2}{\|\alpha_{[k]}\|^2}. \quad (19)$$

Remark 7. *If all data-points x_i are normalized such that $\|x_i\| \leq 1 \forall i \in [n]$, then $\sigma_k \leq |\mathcal{P}_k| = n_k$. Furthermore, if we assume that the data partition is balanced, i.e. that $n_k = n/K$ for all k , then $\sigma \leq n^2/K$.*

This can also be used to bound the constants $R^{(t)}$ appearing above as $R^{(t)} \leq \frac{4L^2 n^2}{K}$.

4.1. Primal-Dual Convergence for General Convex Losses

The following theorem shows the convergence for non-smooth loss functions, in terms of objective values as well as primal-dual gap. The analysis in (Jaggi et al., 2014) only covered the case of smooth loss functions.

Theorem 8. *Consider Algorithm 1 starting with $\alpha^{(0)} = \mathbf{0} \in \mathbb{R}^n$ and let Assumption 1 hold and $\forall i \in [n] : \ell_i(\cdot)$ be L -Lipschitz continuous and $\epsilon_G > 0$ be the desired duality gap (and hence an upper-bound on primal sub-optimality). Then after T iterations, where*

$$T \geq T_0 + \max\left\{\left\lceil \frac{1}{\gamma(1-\Theta)} \right\rceil, \frac{4L^2\sigma\sigma'}{\lambda n^2\epsilon_G\gamma(1-\Theta)}\right\}, \quad (20)$$

$$T_0 \geq t_0 + \left(\frac{2}{\gamma(1-\Theta)} \left(\frac{8L^2\sigma\sigma'}{\lambda n^2\epsilon_G} - 1\right)\right)_+, \quad (21)$$

$$t_0 \geq \max\left(0, \left\lceil \frac{1}{\gamma(1-\Theta)} \log\left(\frac{2\lambda n^2(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)}))}{4L^2\sigma\sigma'}\right) \right\rceil\right), \quad (22)$$

we have that the expected duality gap satisfies

$$\mathbb{E}[\mathcal{P}(w(\bar{\alpha})) - \mathcal{D}(\bar{\alpha})] \leq \epsilon_G,$$

at the averaged iterate

$$\bar{\alpha} := \frac{1}{T-T_0} \sum_{t=T_0+1}^{T-1} \alpha^{(t)}. \quad (23)$$

The following corollary of the above theorem clarifies our main result: The more aggressive adding of the partial updates, as compared averaging, offers a very significant improvement in terms of total iterations needed. While the convergence in the ‘adding’ case becomes independent of the number of machines K , the ‘averaging’ regime shows the known degradation of the rate with growing K , which is a major drawback of the original COCOA algorithm.

This important difference in the convergence speed is not a theoretical artifact but also confirmed in our practical experiments below for different K , as shown e.g. in Figure 2.

We will demonstrate below that by choosing γ and σ' accordingly, we can still recover the original COCOA algorithm and its rate.

Corollary 9. *Assume that all datapoints x_i are bounded as $\|x_i\| \leq 1$ and that the data partition is balanced, i.e. that $n_k = n/K$ for all k . We consider two different possible choices of the aggregation parameter γ :*

- (COCOA Averaging, $\gamma := \frac{1}{K}$): *In this case, $\sigma' := 1$ is a valid choice which satisfies (10). Then using $\sigma \leq n^2/K$ in light of Remark 7, we have that T iterations are sufficient for primal-dual accuracy ϵ_G , with*

$$\begin{aligned} T &\geq T_0 + \max\left\{\left\lceil \frac{K}{(1-\Theta)} \right\rceil, \frac{4L^2}{\lambda\epsilon_G(1-\Theta)}\right\}, \\ T_0 &\geq t_0 + \left(\frac{2K}{(1-\Theta)} \left(\frac{8L^2}{\lambda K\epsilon_G} - 1\right)\right)_+, \\ t_0 &\geq \max(0, \left\lceil \frac{K}{(1-\Theta)} \log\left(\frac{2\lambda(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)}))}{4KL^2}\right) \right\rceil). \end{aligned}$$

Hence the more machines K , the more iterations are needed (in the worst case).

- (COCOA⁺ Adding, $\gamma := 1$): *In this case, the choice of $\sigma' := K$ satisfies (10). Then using $\sigma \leq n^2/K$ in light of Remark 7, we have that T iterations are sufficient for primal-dual accuracy ϵ_G , with*

$$\begin{aligned} T &\geq T_0 + \max\left\{\left\lceil \frac{1}{(1-\Theta)} \right\rceil, \frac{4L^2}{\lambda\epsilon_G(1-\Theta)}\right\}, \\ T_0 &\geq t_0 + \left(\frac{2}{(1-\Theta)} \left(\frac{8L^2}{\lambda\epsilon_G} - 1\right)\right)_+, \\ t_0 &\geq \max(0, \left\lceil \frac{1}{(1-\Theta)} \log\left(\frac{2\lambda n(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)}))}{4L^2 K}\right) \right\rceil) \end{aligned}$$

which is significantly better than in the averaging case above.

In practice, usually $\sigma \ll n^2/K$, and hence the actual convergence rate can be much better than the proven worst-case bound. Table 1 shows that the actual value of σ is typically between one and two orders of magnitudes smaller compared to our used upper-bound n^2/K .

Table 1. The ratio of upper-bound $\frac{n^2}{K}$ divided by the true value of the parameter σ , for some real datasets.

K	16	32	64	128	256	512
news	15.483	14.933	14.278	13.390	12.074	10.252
real-sim	42.127	36.898	30.780	23.814	16.965	11.835
rcv1	40.138	23.827	28.204	21.792	16.339	11.099
K	256	512	1024	2048	4096	8192
covtype	17.277	17.260	17.239	16.948	17.238	12.729

4.2. Primal-Dual Convergence for Smooth Losses

The following theorem shows the convergence for smooth loss functions, in terms of objective values as well as primal-dual gap.

Theorem 10. *Assume the loss functions ℓ_i are $(1/\mu)$ -smooth, for $i \in [n]$. We define $\sigma_{\max} = \max_{k \in [K]} \sigma_k$.*

Then after T iterations of Algorithm 1, with

$$T \geq \frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \frac{1}{\epsilon_{\mathcal{D}}},$$

it holds that

$$\mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(T)})] \leq \epsilon_{\mathcal{D}}.$$

Furthermore, after T iterations with

$$T \geq \frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \left(\frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \frac{1}{\epsilon_G} \right),$$

we have the expected duality gap

$$\mathbb{E}[\mathcal{P}(w(\alpha^{(T)})) - \mathcal{D}(\alpha^{(T)})] \leq \epsilon_G.$$

The following corollary clarifies the impact of the different aggregation of the updates in the COCOA⁺ scheme compared to COCOA, for the case of smooth losses. It will again show that while the COCOA variant suffers from degrading with the increase of the number of machines K , the new COCOA⁺ rate becomes independent of K .

Corollary 11. *Assume that all datapoints x_i are bounded as $\|x_i\| \leq 1$ and that the data partition is balanced, i.e. that $n_k = n/K$ for all k . We again consider the same two different possible choices of the aggregation parameter γ :*

- (COCOA Averaging, $\gamma := \frac{1}{K}$): *In this case, $\sigma' := 1$ is a valid choice which satisfies (10). Then using $\sigma_{\max} \leq n_k = n/K$ in light of Remark 7, we have that T iterations are sufficient for suboptimality $\epsilon_{\mathcal{D}}$, with*

$$T \geq \frac{1}{1-\Theta} \frac{\lambda\mu K + 1}{\lambda\mu} \log \frac{1}{\epsilon_{\mathcal{D}}}$$

Hence the more machines K , the more iterations are needed (in the worst case).

- (CoCoA⁺ Adding, $\gamma := 1$): In this case, the choice of $\sigma' := K$ satisfies (10). Then using $\sigma_{\max} \leq n_k = n/K$ in light of Remark 7, we have that T iterations are sufficient for suboptimality ϵ_G , with

$$T \geq \frac{1}{(1-\Theta)} \frac{\lambda\mu+1}{\lambda\mu} \log \frac{1}{\epsilon_D}$$

which is significantly better than in the averaging case above. Both convergence rates hold analogously for the duality gap.

4.3. Comparison with Original CoCoA

Remark 12. If we choose the averaging option $\gamma := \frac{1}{K}$ for aggregating the updates, together with $\sigma' := 1$, then the resulting Algorithm 1 is identical with CoCoA analyzed in (Jaggi et al., 2014). However, their results provide only convergence for smooth loss functions ℓ_i and provide guarantees for dual sub-optimality and not the duality gap.

Formally, when $\sigma' = 1$, the subproblems (9) will differ from the original dual $\mathcal{D}(\cdot)$ only by an additive constant, which does not affect the local optimization algorithms used within CoCoA.

5. SDCA as a Local Solver

The previous Section 4 has shown the convergence rates of Algorithm 1 provided that one uses a local solver giving an approximation quality Θ as described in Assumption 1. The obvious question to be answered is which solver can provide us a solution of that specified quality.

In this section we will show that coordinate ascent (SDCA) applied on the local subproblem will indeed deliver such a solution. The LOCALSDCA solver is summarized in Algorithm 2. As an input, the methods receives the local α variables, as well as a shared vector $w \stackrel{(3)}{=} w(\alpha)$ being compatible with the last state of all local $\alpha \in \mathbb{R}^n$ variables. The LOCALSDCA algorithm will then produce a random sequence of iterates $\{\Delta\alpha_{[k]}^{(h)}\}_{h=1}^H$, on each part $k \in [K]$.

Algorithm 2 LOCALSDCA ($\alpha_{[k]}, w, k, H$)

- 1: **Input:** $\alpha_{[k]}, w = w(\alpha)$
 - 2: **Data:** Local $\{(x_i, y_i)\}_{i \in \mathcal{P}_k}$
 - 3: **Initialize:** $\Delta\alpha_{[k]}^{(0)} \leftarrow \mathbf{0} \in \mathbb{R}^n$
 - 4: **for** $h = 0, 1, \dots, H - 1$ **do**
 - 5: choose $i \in \mathcal{P}_k$ uniformly at random
 - 6: $\delta_i^* = \arg \max_{\delta_i \in \mathbb{R}} \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h)} + \delta_i e_i, w)$
 - 7: $\Delta\alpha_{[k]}^{(h+1)} \leftarrow \Delta\alpha_{[k]}^{(h)} + \delta_i^* e_i$
 - 8: **end for**
 - 9: **Output:** $\Delta\alpha_{[k]}^{(H)}$
-

The following two Theorems 13, 14 characterize the convergence of the LOCALSDCA method given in Algorithm 2, for both smooth and non-smooth functions.

In all the results we will use $r_{\max} := \max_{i \in [n]} \|x_i\|^2$.

Theorem 13. Assume the functions ℓ_i are $(1/\mu)$ -smooth for $i \in [n]$. Then Assumption 1 on the local approximation quality Θ is satisfied for LOCALSDCA as given in Algorithm 2, if we choose the number of inner iterations H as

$$H \geq n_k \frac{\sigma' r_{\max} + \lambda n \mu}{\lambda n \mu} \log \frac{1}{\Theta}. \quad (24)$$

Theorem 14. Assume the functions ℓ_i are L -Lipschitz for $i \in [n]$. Then Assumption 1 on the local approximation quality Θ is satisfied for LOCALSDCA as given in Algorithm 2, if we choose the number of inner iterations H as

$$H \geq n_k \left(\frac{1 - \Theta}{\Theta} + \frac{\sigma' r_{\max}}{2\Theta \lambda n^2} \frac{\|\Delta\alpha_{[k]}^*\|^2}{\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w)} \right). \quad (25)$$

Remark 15. Between the different regimes allowed in CoCoA⁺ – ranging between averaging and adding the updates – the computational cost for obtaining the required local approximation quality varies with the choice of σ' . From the above worst-case upper bound, we note that the cost can increase with σ' , as aggregation becomes more aggressive. However, as we will see in the practical experiments in Section 7 below, the additional cost is negligible compared to the gain in speed from the different aggregation, when measured on real datasets.

6. Discussion and Related Work

SGD-based Algorithms. For the empirical loss minimization problems of interest here, stochastic subgradient descent (SGD) based methods are well-established. Several distributed variants of SGD have been proposed (Niu et al., 2011; Liu et al., 2014; Duchi et al., 2013). Most of these build on the idea of a parameter-server, receiving all SGD updates performed by each local worker. The downside of this approach is that the amount of required communication is equal to the amount of data read locally (as in mini-batch SGD with a batch size of 1 per worker). Being tailored for very sparse problems, such SGD variants are in practice not competitive with the more communication efficient methods considered here, which allow more local updates per round.

One-Shot Communication Schemes. At the other extreme of the spectrum, we find the group of distributed methods only using a single round of communication such as (Zhang et al., 2013; Zinkevich et al., 2010; Mann et al., 2009; Zhang et al., 2013; McWilliams et al., 2014). These do require additional assumptions on the partitioning of

the data, and furthermore can not guarantee convergence to the optimum solution for all regularizers, as shown e.g. in (Shamir et al., 2014). (Balcan et al., 2012) shows additional relevant lower bounds on the minimum number of communication rounds necessary for a given approximation quality for similar machine learning problems.

Methods Allowing Local Optimization. It is apparent that the well performing methods must lie in the middle of the two extremes of the communication vs computation tradeoff. Therefore, we need to design more meaningful data-local subproblems to be solved per round of communication. In this spirit, (Shamir et al., 2014; Zhang & Xiao, 2015) have proposed distributed Newton-type algorithms. In this approach, the subproblems have to be solved to very high accuracy for the convergence rate to hold, which is often prohibitive as the size of the data on one machine is still relative large.

The CoCoA framework (Jaggi et al., 2014) allows using local solvers of weak local approximation quality in each round, while still giving a convergence rate for smooth losses. By making use of the primal-dual structure in the line of work of (Yu et al., 2012; Shalev-Shwartz & Zhang, 2013c; Yang, 2013; Yang et al., 2013; Jaggi et al., 2014), the CoCoA framework as well as the extension here allow to have more control over the meaningful aggregation of updates between different machines.

Mini-Batch Methods. Mini-batch versions of both SGD and coordinate descent (CD) (Richtárik & Takáč, 2013b;a; Shalev-Shwartz & Zhang, 2013b; Yang, 2013; Qu & Richtárik, 2014; Qu et al., 2014) suffer from their convergence rate degrading towards the rate of batch gradient descent, as the size of the mini-batch is increased. This follows because the updates from all datapoints in the mini-batch are based on the previous parameter vector w , contrasting methods that allow local updates such as CoCoA or local-SGD (Jaggi et al., 2014). Furthermore, the aggregation parameter for mini-batch methods is harder to tune, as it can lie anywhere in the order of mini-batch size, being usually in the thousands in practice. The mini-batch size must be chosen by the tradeoff of cost of a communication round vs. local reading of a vector. In the CoCoA setting, the aggregation parameter is slightly easier to tune, as it is in the smaller range given by number of machines. In our CoCoA⁺ extension presented below, we show that we can get rid of tuning the aggregation parameter completely, by simply allowing additive aggregation.

ADMM. An alternative approach to distributed problems of our form of interest is to use the alternating direction method of multipliers (ADMM), as used for distributed SVM training e.g. in (Forero et al., 2010). This uses a penalty parameter balancing between the equality con-

Table 2. Datasets for Numerical Experiments.

Dataset	n	d	Sparsity
covertypes	522,911	54	22.22%
epsilon	400,000	2,000	100%
RCV1	677,399	47,236	0.16%

straint w and the optimization objective (Boyd et al., 2011). However, the known convergence rates for ADMM are weaker than the more problem tailored methods mentioned previously, and also the choice of the penalty parameter is often unclear.

7. Numerical Experiments

We present experiments on several large real-world datasets distributed across multiple machines, running on Apache Spark. We show that CoCoA⁺ converges to the optimal solution faster in terms of total rounds as well as elapsed time as compared to CoCoA in all cases, despite varying: the dataset, values of regularization, batch size, and cluster size (Section 7.2). In Section 7.3 we demonstrate that this performance translates to orders of magnitude improvement in convergence when scaling up the number of machines K , as compared to CoCoA as well as to several other state-of-the-art methods. Finally, in Section 7.4 we investigate the impact of the subproblem parameter σ' in the CoCoA⁺ framework.

7.1. Implementation Details

We implement CoCoA⁺ and all other algorithms for comparison in Apache Spark (Zaharia et al., 2012), and run them in the Amazon cloud, using m3.large EC2 instances. We apply all methods to the binary hinge-loss support vector machine. The analysis for this non-smooth loss was not covered in (Jaggi et al., 2014) but has been captured here, and thus is both theoretically and practically justified. A summary of the datasets used is shown in Table 2.

7.2. Comparison of CoCoA⁺ and CoCoA

We compare the CoCoA⁺ and CoCoA frameworks directly using two datasets (Covertypes and RCV1) across various values of λ , the regularizer, in Figure 1. For each value of λ we consider both methods with different values of H , the number of local iterations performed before communicating to the master. For all runs of CoCoA⁺ we use the safe upper bound of γK for σ' . In terms of both the total number of communications made and the elapsed time, CoCoA⁺ (shown in blue) converges to the optimal solution faster than CoCoA (red). The discrepancy is larger for greater values of λ , where the strongly convex regularizer has more of an impact and the problem difficulty is reduced. We also see a greater performance gap for smaller

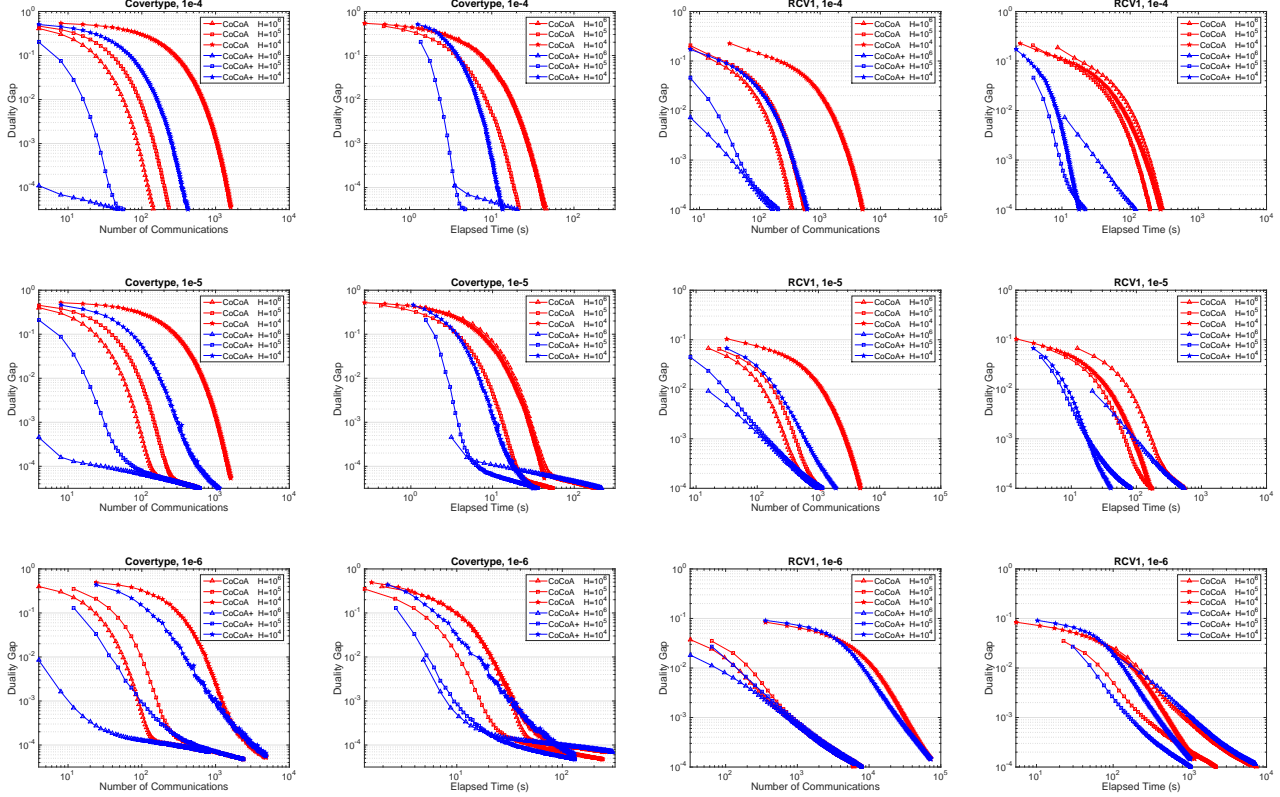


Figure 1. Duality gap vs. the number of communicated vectors, as well as duality gap vs. elapsed time in seconds for two datasets: Covertypes (left, $K=4$) and RCV1 (right, $K=8$). Both are shown on a log-log scale, and for three different values of regularization ($\lambda=1e-4$; $1e-5$; $1e-6$). Each plot contains a comparison of CoCoA (red) and CoCoA⁺ (blue), for three different values of H , the number of local iterations performed per round. For all plots, across all values of λ and H , we see that CoCoA⁺ converges to the optimal solution faster than CoCoA, in terms of both the number of communications and the elapsed time.

values of H , where there is frequent communication between the machines and the master, and changes between the algorithms therefore play a larger role.

7.3. Scaling the Number of Machines K

In Figure 2 we demonstrate the ability of CoCoA⁺ to scale with an increasing number of machines K . The experiments confirm the ability of strong scaling of the new method, as predicted by our theory in Section 4, in contrast to the competing methods. Unlike CoCoA, which becomes linearly slower when increasing the number of machines, the performance of CoCoA⁺ improves with additional machines, only starting to degrade slightly once $K=16$ for the RCV1 dataset.

7.4. Impact of the Subproblem Parameter σ'

Finally, in Figure 3, we consider the effect of the choice of the subproblem parameter σ' on convergence. We plot both the number of communications and clock time on a log-

log scale for the RCV1 dataset with $K=8$ and $H=1e4$. For $\gamma = 1$ (the most aggressive variant of CoCoA⁺ in which updates are added) we consider several different values of σ' , ranging from $\sigma'=1$ to $\sigma'=8$. The value $\sigma'=8$ represents the safe upper bound of γK . We see that the optimal convergence occurs around $\sigma'=4$, and diverges for $\sigma' \leq 2$. Notably, we see that the easy to calculate upper bound of $\sigma' := \gamma K$ (as given by Lemma 3) has only slightly worse performance than best possible subproblem parameter in our setting.

8. Conclusion

In conclusion, we present a novel framework CoCoA⁺ that allows for fast and communication-efficient *additive aggregation* in distributed algorithms for primal-dual optimization. We analyze the theoretical performance of this method, giving strong primal-dual convergence rates with outer iterations scaling independently of the number of machines. We extended our theory to allow for non-smooth losses. Our experimental results show significant speedups over previous methods, including the original CoCoA

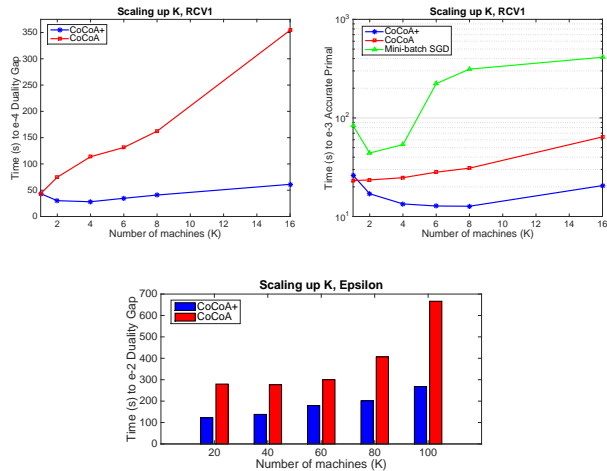


Figure 2. The effect of increasing the number of machines K on the time (s) to reach an ϵ_D -accurate solution. We see that CoCoA⁺ converges twice as fast as CoCoA on 100 machines for the Epsilon dataset, and nearly 7 times as quickly for the RCV1 dataset. Mini-batch SGD converges an order of magnitude more slowly than both methods.

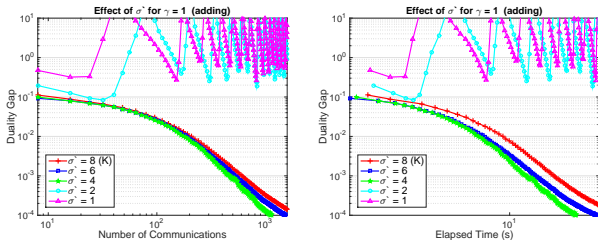


Figure 3. The effect of σ' on convergence of CoCoA⁺ for the RCV1 dataset distributed across $K=8$ machines. Decreasing σ' improves performance in terms of communication and overall run time until a certain point, after which the algorithm diverges. The “safe” upper bound of $\sigma' := K=8$ has only slightly worse performance than the practically best “un-safe” value of σ' .

framework as well as other state-of-the-art methods.

References

Balcan, Maria-Florina, Blum, Avrim, Fine, Shai, and Mansour, Yishay. Distributed Learning, Communication Complexity and Privacy. In *COLT*, pp. 26.1–26.22, April 2012.

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Duchi, John C, Jordan, Michael I, and McMahan, H Bren-

dan. Estimation, Optimization, and Parallelism when Data is Sparse. In *NIPS*, 2013.

Fercoq, Olivier and Richtárik, Peter. Accelerated, parallel and proximal coordinate descent. *arXiv:1312.5799*, 2013.

Fercoq, Olivier, Qu, Zheng, Richtárik, Peter, and Takáč, Martin. Fast distributed coordinate descent for non-strongly convex losses. *IEEE Workshop on Machine Learning for Signal Processing*, 2014, 2014.

Forero, Pedro A, Cano, Alfonso, and Giannakis, Georgios B. Consensus-Based Distributed Support Vector Machines. *JMLR*, 11:1663–1707, 2010.

Jaggi, Martin, Smith, Virginia, Takáč, Martin, Terhorst, Jonathan, Krishnan, Sanjay, Hofmann, Thomas, and Jordan, Michael I. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pp. 3068–3076, 2014.

Konečný, Jakub, Ma, Chenxin, Richtárik, Peter, Jaggi, Martin, and Takáč, Martin. Distributed optimization with arbitrary local solvers. Technical report, 2014.

Liu, Ji and Wright, Stephen J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *arXiv:1403.3862*, 2014.

Liu, Ji, Wright, Stephen J, Ré, Christopher, Bittorf, Victor, and Sridhar, Srikrishna. An Asynchronous Parallel Stochastic Coordinate Descent Algorithm. In *ICML*, 2014.

Lu, Zhaosong and Xiao, Lin. On the complexity analysis of randomized block-coordinate descent methods. *arXiv preprint arXiv:1305.4723*, 2013.

Mann, Gideon, McDonald, Ryan, Mohri, Mehryar, Silberman, Nathan, and Walker, Daniel D. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. *NIPS*, pp. 1231–1239, 2009.

Mareček, Jakub, Richtárik, Peter, and Takáč, Martin. Distributed block coordinate descent for minimizing partially separable functions. *arXiv:1406.0238*, 2014.

McWilliams, Brian, Heinze, Christina, Meinshausen, Nicolai, Krummenacher, Gabriel, and Vanchinathan, Hastagiri P. LOCO: Distributing Ridge Regression with Random Projections. In *NIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*, June 2014.

Niu, Feng, Recht, Benjamin, Ré, Christopher, and Wright, Stephen J. Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *NIPS*, 2011.

- Qu, Zheng and Richtárik, Peter. Coordinate descent with arbitrary sampling I: Algorithms and complexity. *arXiv:1412.8060*, 2014.
- Qu, Zheng, Richtárik, Peter, and Zhang, Tong. Randomized dual coordinate ascent with arbitrary sampling. *arXiv:1411.5873*, 2014.
- Richtárik, Peter and Takáč, Martin. Parallel coordinate descent methods for big data optimization. *arXiv:1212.0873*, 2012.
- Richtárik, Peter and Takáč, Martin. On optimal probabilities in stochastic coordinate descent methods. *arXiv:1310.3438*, 2013a.
- Richtárik, Peter and Takáč, Martin. Distributed coordinate descent method for learning with big data. *arXiv preprint arXiv:1310.2059*, 2013b.
- Richtárik, Peter and Takáč, Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, April 2014.
- Shalev-Shwartz, Shai and Zhang, Tong. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS*, pp. 378–385, 2013a.
- Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *arXiv:1309.2375*, 2013b.
- Shalev-Shwartz, Shai and Zhang, Tong. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *JMLR*, 14:567–599, February 2013c.
- Shamir, Ohad and Srebro, Nathan. Distributed Stochastic Optimization and Learning . In *Allerton*, 2014.
- Shamir, Ohad, Srebro, Nathan, and Zhang, Tong. Communication efficient distributed optimization using an approximate newton-type method. In *ICML*, 2014.
- Tappenden, Rachael, Takáč, Martin, and Richtárik, Peter. On the complexity of parallel coordinate descent. Technical report, 2015. ERGO 15-001, University of Edinburgh.
- Yang, Tianbao. Trading Computation for Communication: Distributed Stochastic Dual Coordinate Ascent. In *NIPS*, 2013.
- Yang, Tianbao, Zhu, Shenghuo, Jin, Rong, and Lin, Yuanqing. On Theoretical Analysis of Distributed Stochastic Dual Coordinate Ascent. *arXiv:1312.1031*, December 2013.
- Yu, Hsiang-Fu, Hsieh, Cho-Jui, Chang, Kai-Wei, and Lin, Chih-Jen. Large Linear Classification When Data Cannot Fit in Memory. *ACM Transactions on Knowledge Discovery from Data*, 5(4):1–23, 2012.
- Zaharia, Matei, Chowdhury, Mosharaf, Das, Tathagata, Dave, Ankur, McCauley, Murphy, Franklin, Michael J, Shenker, Scott, and Stoica, Ion. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In *NSDI 2012*, 2012.
- Zhang, Yuchen and Xiao, Lin. Communication-efficient distributed optimization of self-concordant empirical loss. *arXiv:1501.00263*, 2015.
- Zhang, Yuchen, Duchi, John C, and Wainwright, Martin J. Communication-Efficient Algorithms for Statistical Optimization. *JMLR*, 14:3321–3363, November 2013.
- Zinkevich, Martin A, Weimer, Markus, Smola, Alex J, and Li, Lihong. Parallelized Stochastic Gradient Descent. *NIPS 2010: Advances in Neural Information Processing Systems 23*, pp. 1–37, 2010.

Part I

Appendix

A. Technical Lemmas

Lemma 16 (Lemma 21 in (Shalev-Shwartz & Zhang, 2013c)). *Let $\ell_i : \mathbb{R} \rightarrow \mathbb{R}$ be an L -Lipschitz continuous. Then for any real value a with $|a| > L$ we have that $\ell_i^*(a) = \infty$.*

Lemma 17. *Assuming the loss functions ℓ_i are bounded by $\ell_i(0) \leq 1$ for all $i \in [n]$ (as we have assumed in (5) above), then for the zero vector $\alpha^{(0)} := \mathbf{0} \in \mathbb{R}^n$, we have*

$$\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)}) = \mathcal{D}(\alpha^*) - \mathcal{D}(\mathbf{0}) \leq 1. \quad (26)$$

Proof. For $\alpha := \mathbf{0} \in \mathbb{R}^n$, we have $w(\alpha) = \frac{1}{\lambda n} A \alpha = \mathbf{0} \in \mathbb{R}^d$. Therefore, by definition of the dual objective \mathcal{D} given in (2),

$$0 \leq \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha) \leq P(w(\alpha)) - \mathcal{D}(\alpha) = 0 - \mathcal{D}(\alpha) \stackrel{(5),(2)}{\leq} 1. \quad \square$$

B. Proofs

B.1. Proof of Lemma 3

See (Richtárik & Takáč, 2013b).

B.2. Proof of Lemma 4

Indeed, we have

$$\mathcal{D}(\alpha + \gamma \sum_{k=1}^K \Delta \alpha_{[k]}) = \underbrace{-\frac{1}{n} \sum_{i=1}^n \ell_i^*(-\alpha_i - \gamma (\sum_{k=1}^K \Delta \alpha_{[k]})_i)}_A - \underbrace{\frac{\lambda}{2} \left\| \frac{1}{\lambda n} A (\alpha + \gamma \sum_{k=1}^K \Delta \alpha_{[k]}) \right\|^2}_B. \quad (27)$$

Now, let us bound the terms A and B separately. We have

$$\begin{aligned} A &= -\frac{1}{n} \sum_{k=1}^K \left(\sum_{i \in \mathcal{P}_k} \ell_i^*(-\alpha_i - \gamma (\Delta \alpha_{[k]})_i) \right) = -\frac{1}{n} \sum_{k=1}^K \left(\sum_{i \in \mathcal{P}_k} \ell_i^*(-(1-\gamma)\alpha_i - \gamma(\alpha + \Delta \alpha_{[k]})_i) \right) \\ &\geq -\frac{1}{n} \sum_{k=1}^K \left(\sum_{i \in \mathcal{P}_k} (1-\gamma)\ell_i^*(-\alpha_i) + \gamma \ell_i^*(-(\alpha + \Delta \alpha_{[k]})_i) \right). \end{aligned}$$

Where the last inequality is due to Jensen. Now we will bound B .

$$\begin{aligned} B &= \left\| \frac{1}{\lambda n} A (\alpha + \gamma \sum_{k=1}^K \Delta \alpha_{[k]}) \right\|^2 = \left\| w(\alpha) + \gamma \frac{1}{\lambda n} \sum_{k=1}^K A \Delta \alpha_{[k]} \right\|^2 = \|w(\alpha)\|^2 + \sum_{k=1}^K 2\gamma \frac{1}{\lambda n} w(\alpha)^T A \Delta \alpha_{[k]} \\ &\quad + \gamma \left(\frac{1}{\lambda n} \right)^2 \gamma \left\| \sum_{k=1}^K A \Delta \alpha_{[k]} \right\|^2 \stackrel{(10)}{\leq} \|w(\alpha)\|^2 + \sum_{k=1}^K 2\gamma \frac{1}{\lambda n} w(\alpha)^T A \Delta \alpha_{[k]} + \gamma \left(\frac{1}{\lambda n} \right)^2 \sigma' \sum_{k=1}^K \|A \Delta \alpha_{[k]}\|^2. \end{aligned}$$

Plugging A and B into (27) will give us

$$\begin{aligned}
 \mathcal{D}(\alpha + \gamma \sum_{k=1}^K \Delta\alpha_{[k]}) &= -\frac{1}{n} \sum_{k=1}^K \left(\sum_{i \in \mathcal{P}_k} (1-\gamma) \ell_i^*(-\alpha_i) + \gamma \ell_i^*(-(\alpha + \Delta\alpha_{[k]})_i) \right) \\
 &\quad - \gamma \frac{\lambda}{2} \|w(\alpha)\|^2 - (1-\gamma) \frac{\lambda}{2} \|w(\alpha)\|^2 - \frac{\lambda}{2} \sum_{k=1}^K 2\gamma \frac{1}{\lambda n} w(\alpha)^T A \Delta\alpha_{[k]} - \frac{\lambda}{2} \gamma \left(\frac{1}{\lambda n}\right)^2 \sigma' \sum_{k=1}^K \|A\alpha_{[k]}\|^2 \\
 &= -\frac{1}{n} \sum_{k=1}^K \underbrace{\left(\sum_{i \in \mathcal{P}_k} (1-\gamma) \ell_i^*(-\alpha_i) \right)}_{(1-\gamma)\mathcal{D}(\alpha)} - (1-\gamma) \frac{\lambda}{2} \|w(\alpha)\|^2 \\
 &\quad + \gamma \sum_{k=1}^K \left(-\frac{1}{n} \sum_{i \in \mathcal{P}_k} \ell_i^*(-(\alpha + \Delta\alpha_{[k]})_i) - \frac{1}{K} \frac{\lambda}{2} \|w(\alpha)\|^2 - \frac{1}{n} w(\alpha)^T A \Delta\alpha_{[k]} - \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A\alpha_{[k]} \right\|^2 \right) \\
 &\stackrel{(9)}{=} (1-\gamma)\mathcal{D}(\alpha) + \gamma \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w).
 \end{aligned}$$

B.3. Proof of Lemma 5

For sake of notation we will write α instead of $\alpha^{(t)}$, w instead of $w(\alpha^{(t)})$ and u instead of $u^{(t)}$.

Now, let us estimate the expected change of the dual objective. Using the definition of the dual update $\alpha^{(t+1)} := \alpha^{(t)} + \gamma \sum_k \Delta\alpha_{[k]}$ resulting in Algorithm 1, we have

$$\begin{aligned}
 \mathbb{E}[\mathcal{D}(\alpha^{(t)}) - \mathcal{D}(\alpha^{(t+1)})] &= \mathbb{E}[\mathcal{D}(\alpha) - \mathcal{D}(\alpha + \gamma \sum_{k=1}^K \Delta\alpha_{[k]})] \\
 &\stackrel{(11)}{\leq} \mathbb{E}[\mathcal{D}(\alpha) - (1-\gamma)\mathcal{D}(\alpha) - \gamma \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w)] = \gamma \mathbb{E}[\mathcal{D}(\alpha) - \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w)] \\
 &= \gamma \mathbb{E}[\mathcal{D}(\alpha) - \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) + \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(t)}, w)] \\
 &\stackrel{(12)}{\leq} \gamma \left(\mathcal{D}(\alpha) - \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) + \Theta \left(\sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \underbrace{\sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\mathbf{0}, w)}_{\mathcal{D}(\alpha)} \right) \right) \\
 &= \gamma(1-\Theta) \left(\underbrace{\mathcal{D}(\alpha) - \sum_{k=1}^K \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w)}_C \right). \tag{28}
 \end{aligned}$$

Now, let us upper bound the C term (we will denote by $\Delta\alpha^* = \sum_{k=1}^K \Delta\alpha_{[k]}^*$):

$$\begin{aligned}
 C &\stackrel{(2),(9)}{=} \frac{1}{n} \sum_{i=1}^n (\ell_i^*(-\alpha_i - \Delta\alpha_i^*) - \ell_i^*(-\alpha_i)) + \frac{1}{n} w(\alpha)^T A \Delta\alpha^* + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A \Delta\alpha_{[k]}^* \right\|^2 \\
 &\leq \frac{1}{n} \sum_{i=1}^n (\ell_i^*(-\alpha_i - s(u_i - \alpha_i)) - \ell_i^*(-\alpha_i)) + \frac{1}{n} w(\alpha)^T A s(u - \alpha) + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A s(u - \alpha)_{[k]} \right\|^2 \\
 &\stackrel{\text{Strong conv.}}{\leq} \frac{1}{n} \sum_{i=1}^n \left(s\ell_i^*(-u_i) + (1-s)\ell_i^*(-\alpha_i) - \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 - \ell_i^*(-\alpha_i) \right) + \frac{1}{n} w(\alpha)^T A s(u - \alpha) \\
 &\quad + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A s(u - \alpha)_{[k]} \right\|^2 \\
 &= \frac{1}{n} \sum_{i=1}^n \left(s\ell_i^*(-u_i) - s\ell_i^*(-\alpha_i) - \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \right) + \frac{1}{n} w(\alpha)^T A s(u - \alpha) + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A s(u - \alpha)_{[k]} \right\|^2.
 \end{aligned}$$

The convex conjugate maximal property implies that

$$\ell_i^*(-u_i) = -u_i w(\alpha)^T x_i - \ell_i(w(\alpha)^T x_i). \quad (29)$$

Moreover, from the definition of the primal and dual optimization problems (1), (2), we can write the duality gap as

$$G(\alpha) := \mathcal{P}(w(\alpha)) - \mathcal{D}(\alpha) \stackrel{(1),(2)}{=} \frac{1}{n} \sum_{i=1}^n (\ell_i(x_i^T w) + \ell_i^*(-\alpha_i) + w(\alpha)^T x_i \alpha_i). \quad (30)$$

Hence,

$$\begin{aligned}
 C &\stackrel{(29)}{\leq} \frac{1}{n} \sum_{i=1}^n \left(-su_i w(\alpha)^T x_i - s\ell_i(w(\alpha)^T x_i) - s\ell_i^*(-\alpha_i) - \underbrace{sw(\alpha)^T x_i \alpha_i + sw(\alpha)^T x_i \alpha_i}_{0} - \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \right) \\
 &\quad + \frac{1}{n} w(\alpha)^T A s(u - \alpha) + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A s(u - \alpha)_{[k]} \right\|^2 \\
 &= \frac{1}{n} \sum_{i=1}^n (-s\ell_i(w(\alpha)^T x_i) - s\ell_i^*(-\alpha_i) - sw(\alpha)^T x_i \alpha_i) + \frac{1}{n} \sum_{i=1}^n \left(sw(\alpha)^T x_i (\alpha_i - u_i) - \frac{\mu}{2}(1-s)s(u_i - \alpha_i)^2 \right) \\
 &\quad + \frac{1}{n} w(\alpha)^T A s(u - \alpha) + \sum_{k=1}^K \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A s(u - \alpha)_{[k]} \right\|^2 \\
 &\stackrel{(30)}{=} -sG(\alpha) - \frac{\mu}{2}(1-s)s \frac{1}{n} \sum_{i=1}^n \|u - \alpha\|^2 + \frac{\sigma'}{2\lambda} \left(\frac{s}{n}\right)^2 \sum_{k=1}^K \|A(u - \alpha)_{[k]}\|^2. \quad (31)
 \end{aligned}$$

Now (15) follows by plugging (31) to (28).

B.4. Proof of Lemma 6

For general convex functions, the strong convexity parameter is $\mu = 0$, and hence the definition of $R^{(t)}$ becomes

$$R^{(t)} \stackrel{(16)}{=} \sum_{k=1}^K \|A(u^{(t)} - \alpha^{(t)})_{[k]}\|^2 \stackrel{(19)}{\leq} \sum_{k=1}^K \sigma_k \| (u^{(t)} - \alpha^{(t)})_{[k]} \|^2 \stackrel{\text{Lemma 16}}{\leq} \sum_{k=1}^K \sigma_k |\mathcal{P}_k| 4L^2.$$

B.5. Proof of Theorem 8

At first let us estimate expected change of dual feasibility. By using the main Lemma 5, we have

$$\begin{aligned}
 \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t+1)})] &= \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t+1)}) + \mathcal{D}(\alpha^{(t)}) - \mathcal{D}(\alpha^{(t)})] \\
 &\stackrel{(15)}{=} \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}) - \gamma(1 - \Theta)sG(\alpha^{(t)}) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 R^{(t)} \\
 &\stackrel{(4)}{=} \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}) - \gamma(1 - \Theta)s(\mathcal{P}(w(\alpha^{(t)})) - \mathcal{D}(\alpha^{(t)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 R^{(t)} \\
 &\leq \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}) - \gamma(1 - \Theta)s(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 R^{(t)} \\
 &\stackrel{(18)}{\leq} (1 - \gamma(1 - \Theta)s)(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 4L^2\sigma. \tag{32}
 \end{aligned}$$

Using (32) recursively we have

$$\begin{aligned}
 \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})] &= (1 - \gamma(1 - \Theta)s)^t (\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 4L^2\sigma \sum_{j=0}^{t-1} (1 - \gamma(1 - \Theta)s)^j \\
 &= (1 - \gamma(1 - \Theta)s)^t (\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 4L^2\sigma \frac{1 - (1 - \gamma(1 - \Theta)s)^t}{\gamma(1 - \Theta)s} \\
 &\leq (1 - \gamma(1 - \Theta)s)^t (\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)})) + s \frac{4L^2\sigma\sigma'}{2\lambda n^2}. \tag{33}
 \end{aligned}$$

Choice of $s = 1$ and $t = t_0 := \max\{0, \lceil \frac{1}{\gamma(1-\Theta)} \log(2\lambda n^2(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)})) / (4L^2\sigma\sigma')) \rceil\}$ will lead to

$$\mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})] \leq (1 - \gamma(1 - \Theta))^{t_0} (\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(0)})) + \frac{4L^2\sigma\sigma'}{2\lambda n^2} \leq \frac{4L^2\sigma\sigma'}{2\lambda n^2} + \frac{4L^2\sigma\sigma'}{2\lambda n^2} = \frac{4L^2\sigma\sigma'}{\lambda n^2}. \tag{34}$$

Now, we are going to show that

$$\forall t \geq t_0 : \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})] \leq \frac{4L^2\sigma\sigma'}{\lambda n^2(1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0))}. \tag{35}$$

Clearly, (34) implies that (35) holds for $t = t_0$. Now imagine that it holds for any $t \geq t_0$ then we show that it also has to hold for $t + 1$. Indeed, using

$$s = \frac{1}{1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0)} \in [0, 1] \tag{36}$$

we obtain

$$\begin{aligned}
 \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t+1)})] &\stackrel{(32)}{\leq} (1 - \gamma(1 - \Theta)s)(\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})) + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 4L^2\sigma \\
 &\stackrel{(35)}{\leq} (1 - \gamma(1 - \Theta)s) \frac{4L^2\sigma\sigma'}{\lambda n^2(1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0))} + \gamma(1 - \Theta)\frac{\sigma'}{2\lambda}\left(\frac{s}{n}\right)^2 4L^2\sigma \\
 &\stackrel{(36)}{=} \frac{4L^2\sigma\sigma'}{\lambda n^2} \left(\frac{1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0) - \gamma(1 - \Theta) + \gamma(1 - \Theta)\frac{1}{2}}{(1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0))^2} \right) \\
 &= \frac{4L^2\sigma\sigma'}{\lambda n^2} \underbrace{\left(\frac{1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0) - \frac{1}{2}\gamma(1 - \Theta)}{(1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0))^2} \right)}_D.
 \end{aligned}$$

Now, we will upperbound D as follows

$$\begin{aligned}
 D &= \frac{1}{1 + \frac{1}{2}\gamma(1 - \Theta)(t + 1 - t_0)} \underbrace{\frac{(1 + \frac{1}{2}\gamma(1 - \Theta)(t + 1 - t_0))(1 + \frac{1}{2}\gamma(1 - \Theta)(t - 1 - t_0))}{(1 + \frac{1}{2}\gamma(1 - \Theta)(t - t_0))^2}}_{\leq 1} \\
 &\leq \frac{1}{1 + \frac{1}{2}\gamma(1 - \Theta)(t + 1 - t_0)},
 \end{aligned}$$

where in the last inequality we have used the fact that geometric mean is less or equal to arithmetic mean.

If $\bar{\alpha}$ is defined as (23) then we obtain that

$$\begin{aligned}
 \mathbb{E}[G(\bar{\alpha})] &= \mathbb{E} \left[G \left(\sum_{t=T_0}^{T-1} \frac{1}{T-T_0} \alpha^{(t)} \right) \right] \leq \frac{1}{T-T_0} \mathbb{E} \left[\sum_{t=T_0}^{T-1} G(\alpha^{(t)}) \right] \\
 &\stackrel{(15),(18)}{\leq} \frac{1}{T-T_0} \mathbb{E} \left[\sum_{t=T_0}^{T-1} \left(\frac{1}{\gamma(1-\Theta)s} (\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^{(t)})) + \frac{4L^2\sigma\sigma's}{2\lambda n^2} \right) \right] \\
 &= \frac{1}{\gamma(1-\Theta)s} \frac{1}{T-T_0} \mathbb{E} \left[\mathcal{D}(\alpha^{(T)}) - \mathcal{D}(\alpha^{(T_0)}) \right] + \frac{4L^2\sigma\sigma's}{2\lambda n^2} \\
 &\leq \frac{1}{\gamma(1-\Theta)s} \frac{1}{T-T_0} \mathbb{E} \left[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(T_0)}) \right] + \frac{4L^2\sigma\sigma's}{2\lambda n^2}. \tag{37}
 \end{aligned}$$

Now, if $T \geq \lceil \frac{1}{\gamma(1-\Theta)} \rceil + T_0$ such that $T_0 \geq t_0$ we obtain

$$\begin{aligned}
 \mathbb{E}[G(\bar{\alpha})] &\stackrel{(37),(35)}{\leq} \frac{1}{\gamma(1-\Theta)s} \frac{1}{T-T_0} \left(\frac{4L^2\sigma\sigma'}{\lambda n^2(1 + \frac{1}{2}\gamma(1-\Theta)(T_0 - t_0))} \right) + \frac{4L^2\sigma\sigma's}{2\lambda n^2} \\
 &= \frac{4L^2\sigma\sigma'}{\lambda n^2} \left(\frac{1}{\gamma(1-\Theta)s} \frac{1}{T-T_0} \frac{1}{1 + \frac{1}{2}\gamma(1-\Theta)(T_0 - t_0)} + \frac{s}{2} \right). \tag{38}
 \end{aligned}$$

Choosing

$$s = \frac{1}{(T - T_0)\gamma(1 - \Theta)} \in [0, 1] \tag{39}$$

gives us

$$\mathbb{E}[G(\bar{\alpha})] \stackrel{(38),(39)}{\leq} \frac{4L^2\sigma\sigma'}{\lambda n^2} \left(\frac{1}{1 + \frac{1}{2}\gamma(1-\Theta)(T_0 - t_0)} + \frac{1}{(T - T_0)\gamma(1-\Theta)} \frac{1}{2} \right). \tag{40}$$

To have right hand side of (40) smaller than ϵ_G it is sufficient to choose T_0 and T such that

$$\frac{4L^2\sigma\sigma'}{\lambda n^2} \left(\frac{1}{1 + \frac{1}{2}\gamma(1-\Theta)(T_0 - t_0)} \right) \leq \frac{1}{2}\epsilon_G, \tag{41}$$

$$\frac{4L^2\sigma\sigma'}{\lambda n^2} \left(\frac{1}{(T - T_0)\gamma(1-\Theta)} \frac{1}{2} \right) \leq \frac{1}{2}\epsilon_G. \tag{42}$$

Hence of if

$$\begin{aligned}
 t_0 + \frac{2}{\gamma(1-\Theta)} \left(\frac{8L^2\sigma\sigma'}{\lambda n^2\epsilon_G} - 1 \right) &\leq T_0, \\
 T_0 + \frac{4L^2\sigma\sigma'}{\lambda n^2\epsilon_G\gamma(1-\Theta)} &\leq T,
 \end{aligned}$$

then (41) and (42) are satisfied.

B.6. Proof of Theorem 10

If the function $\ell_i(\cdot)$ is $(1/\mu)$ -smooth then $\ell_i^*(\cdot)$ is μ -strongly convex with respect to the $\|\cdot\|$ norm. From (16) we have

$$\begin{aligned}
 R^{(t)} &\stackrel{(16)}{=} -\frac{\lambda\mu n(1-s)}{\sigma's} \|u^{(t)} - \alpha^{(t)}\|^2 + \sum_{k=1}^K \|A(u^{(t)} - \alpha^{(t)})_{[k]}\|^2 \\
 &\stackrel{(19)}{\leq} -\frac{\lambda\mu n(1-s)}{\sigma's} \|u^{(t)} - \alpha^{(t)}\|^2 + \sum_{k=1}^K \sigma_k \|u^{(t)} - \alpha_{[k]}^{(t)}\|^2 \\
 &\leq -\frac{\lambda\mu n(1-s)}{\sigma's} \|u^{(t)} - \alpha^{(t)}\|^2 + \sigma_{\max} \sum_{k=1}^K \|u^{(t)} - \alpha_{[k]}^{(t)}\|^2 \\
 &= \left(-\frac{\lambda\mu n(1-s)}{\sigma's} + \sigma_{\max} \right) \|u^{(t)} - \alpha^{(t)}\|^2. \tag{43}
 \end{aligned}$$

If we plug

$$s = \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} \in [0, 1] \quad (44)$$

into (43) we obtain that $\forall t : R^{(t)} \leq 0$. Putting the same s into (15) will give us

$$\mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^{(t)})] \stackrel{(15),(44)}{\geq} \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} G(\alpha^{(t)}) \geq \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}). \quad (45)$$

Using the fact that $\mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^{(t)})] = \mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^*)] + \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})$ we have

$$\mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^*)] + \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}) \stackrel{(45)}{\geq} \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})$$

which is equivalent with

$$\mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t+1)})] \leq \left(1 - \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'}\right) \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)}). \quad (46)$$

Therefore if we denote by $\epsilon_{\mathcal{D}}^{(t)} = \mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})$ we have that

$$\mathbb{E}[\epsilon_{\mathcal{D}}^{(t)}] \stackrel{(46)}{\leq} \left(1 - \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'}\right)^t \epsilon_{\mathcal{D}}^{(0)} \stackrel{(26)}{\leq} \left(1 - \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'}\right)^t \leq \exp\left(-t\gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'}\right).$$

The right hand side will be smaller than some $\epsilon_{\mathcal{D}}$ if

$$t \geq \frac{1}{\gamma(1 - \Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \frac{1}{\epsilon_{\mathcal{D}}}.$$

Moreover, to bound the duality gap, we have

$$\gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} G(\alpha^{(t)}) \stackrel{(45)}{\leq} \mathbb{E}[\mathcal{D}(\alpha^{(t+1)}) - \mathcal{D}(\alpha^{(t)})] \leq \mathbb{E}[\mathcal{D}(\alpha^*) - \mathcal{D}(\alpha^{(t)})].$$

Therefore $G(\alpha^{(t)}) \leq \frac{1}{\gamma(1 - \Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \epsilon_{\mathcal{D}}^{(t)}$. Hence if $\epsilon_{\mathcal{D}} \leq \gamma(1 - \Theta) \frac{\lambda\mu n}{\lambda\mu n + \sigma_{\max}\sigma'} \epsilon_G$ then $G(\alpha^{(t)}) \leq \epsilon_G$. Therefore after

$$t \geq \frac{1}{\gamma(1 - \Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \left(\frac{1}{\gamma(1 - \Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \frac{1}{\epsilon_G} \right)$$

iterations we have obtained a duality gap less than ϵ_G .

B.7. Proof of Theorem 13

Because ℓ_i are $(1/\mu)$ -smooth then functions ℓ_i^* are μ strongly convex with respect to the norm $\|\cdot\|$. The proof is based on techniques developed in recent coordinate descent papers, including (Richtárik & Takáč, 2014; 2013b; 2012; Tappenden et al., 2015; Mareček et al., 2014; Fercoq & Richtárik, 2013; Lu & Xiao, 2013; Fercoq et al., 2014; Qu & Richtárik, 2014; Qu et al., 2014) (Efficient accelerated variants were considered in (Fercoq & Richtárik, 2013; Shalev-Shwartz & Zhang, 2013a)).

First, let us define the function $F(\zeta) : \mathbb{R}^{n_k} \rightarrow \mathbb{R}$ as $F(\zeta) := -\mathcal{G}_k^{\sigma'}(\sum_{i \in \mathcal{P}_k} \zeta_i e_i, w)$. This function can be written in two parts $F(\zeta) = \Phi(\zeta) + f(\zeta)$. The first part denoted by $\Phi(\zeta) = \frac{1}{n} \sum_{i \in \mathcal{P}_k} \ell_i^*(-\alpha_i - \zeta_i)$ is strongly convex with convexity parameter $\frac{\mu}{n}$ with respect to the standard Euclidean norm. In our application, we think of the ζ variable collecting the local dual variables $\Delta\alpha_{[k]}$.

The second part we will denote by $f(\zeta) = \frac{1}{K} \frac{\lambda}{2} \|w(\alpha)\|^2 + \frac{1}{n} \sum_{i \in \mathcal{P}_k} w(\alpha)^T x_i \zeta_i + \frac{\lambda}{2} \sigma' \frac{1}{\lambda^2 n^2} \|\sum_{i \in \mathcal{P}_k} x_i \zeta_i\|^2$. It is easy to show that the gradient of f is coordinate-wise Lipschitz continuous with Lipschitz constant $\frac{\sigma'}{\lambda n^2} r_{\max}$ with respect to the standard Euclidean norm.

Following the proof of Theorem 20 in (Richtárik & Takáč, 2012), we obtain that

$$\begin{aligned} \mathbb{E}[\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h+1)}, w)] &\leq \left(1 - \frac{1}{n_k} \frac{1 + \frac{\mu n \lambda}{\sigma' r_{\max}}}{\frac{\mu n \lambda}{\sigma' r_{\max}}}\right) \left(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h)}, w)\right) \\ &= \left(1 - \frac{1}{n_k} \frac{\lambda n \mu}{\sigma' r_{\max} + \lambda n \mu}\right) \left(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h)}, w)\right). \end{aligned}$$

Over all steps up to step h , this gives

$$\mathbb{E}[\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h)}, w)] \leq \left(1 - \frac{1}{n_k} \frac{\lambda n \mu}{\sigma' r_{\max} + \lambda n \mu}\right)^h \left(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w)\right).$$

Therefore, choosing H as in the assumption of our Theorem, given in Equation (24), we are guaranteed that $\left(1 - \frac{1}{n_k} \frac{\lambda n \mu}{\sigma' r_{\max} + \lambda n \mu}\right)^H \leq \Theta$, as desired.

B.8. Proof of Theorem 14

Similarly as in the proof of Theorem 13 we define a composite function $F(\zeta) = f(\zeta) + \Phi(\zeta)$. However, in this case functions ℓ_i^* are not guaranteed to be strongly convex. However, the first part has still a coordinate-wise Lipschitz continuous gradient with constant $\frac{\sigma'}{\lambda n^2} r_{\max}$ with respect to the standard Euclidean norm. Therefore from Theorem 3 in (Tappenden et al., 2015) we have that

$$\mathbb{E}[\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{(h)}, w)] \leq \frac{n_k}{n_k + h} \left(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w) + \frac{1}{2} \frac{\sigma' r_{\max}}{\lambda n^2} \|\Delta\alpha_{[k]}^*\|^2\right). \quad (47)$$

Now, choice of $h = H$ from (25) is sufficient to have the right hand side of (47) to be $\leq \Theta(\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^*, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w))$.