ON CONVERGENCE ANALYSIS OF GRADIENT BASED PRIMAL-DUAL METHOD OF MULTIPLIERS

Guoqiang Zhang[1]*, Matthew O'Connor*[2]*, and Le Li*[3]

[1]University of Technology Sydney, Australia
[2]Australian National University, Australia
[3]Northwestern Polytechnical University, China

ABSTRACT

Recently, the primal-dual method of multipliers (PDMM) has been proposed and successfully applied to solve a number of decomposable convex optimizations distributedly and iteratively. In this work, we study the gradient based PDMM (GPDMM), where the objective functions are approximated using the gradient information per iteration. It is shown that for a certain class of decomposable convex optimizations, synchronous GPDMM has a sublinear convergence rate of $\mathcal{O}(1/K)$ (where K denotes the iteration index). Experiments on a problem of distributed ridge regularized logistic regression demonstrate the efficiency of synchronous GPDMM.

Index Terms— Distributed optimization, ADMM, PDMM, convergence analysis

1. INTRODUCTION

In recent years distributed optimization has attracted increasing attention driven by two main motivations. Firstly, various types of networks have been proposed and employed for collecting data, monitoring the environment, and managing facilities such as wireless sensor networks, smart grid and Internet of things. In the above situation, distributed optimization is desirable to perform distributed signal processing, network resource allocation, and utility maximization [1]. Secondly, processing of big data usually requires many computing units (e.g., a computer or a GPU) to work jointly, where each unit processes a portion of the data. Distributed optimization is then required for coordination among the computing units [2].

One approach in designing the distributed methods is to treat or reformulate an optimization problem as performing inference over a probabilistic graphical model see ([3–5]). Another approach is to treat or reformulate an optimization problem as a decomposable optimization over a graphical model directly instead of relying on probability theory. Various methods have been proposed in the literature for solving decomposable convex optimizations, such as the dual-averaging algorithm [6], the subgradient algorithm [7], the diffusion adaptation algorithm [8], the exact first-order algorithm (EXTRA) [9], and the alternating direction method of multipliers (ADMM) [10]. In addition, several research attempts have been made to tackle decomposable nonconvex optimizations (see [11–13]).

Due to the generality and effectiveness of ADMM, extensive studies have been conducted for the method in the last few years. The research activities on ADMM can be roughly classified as convergence rate analysis (e.g., [14, 15]), computational simplifications by using gradient information [16, 17], and its applications to real world problems [10]. It is analyzed that the effectiveness of ADMM might be due to the fact that the method belongs to the primal-dual approaches [18, 19], which attempt to solve both the primal and dual problems simultaneously.

Recently, we have proposed the primal-dual method of multipliers (PDMM) [20, 21] for solving a general class of decomposable convex optimizations over a graphical model $G = \{\mathcal{V}, \mathcal{E}\}$, which takes the form:

$$\min \sum_{i\in\mathcal{V}} f_i(\boldsymbol{x}_i) \text{ s.t. } \boldsymbol{A}_{i|j}\boldsymbol{x}_i + \boldsymbol{A}_{j|i}\boldsymbol{x}_j = \boldsymbol{c}_{ij}\ \forall (i,j) \in \mathcal{E}, \quad (1)$$

where s.t. stands for "subject to", each function f_i is assumed to be closed, proper and convex, and the two matrices $(\boldsymbol{A}_{i|j}, \boldsymbol{A}_{j|i})$ and the vector $\boldsymbol{c}_{ij}$ are known a priori for each edge $(i,j) \in \mathcal{E}$. The above formulation (1) enforces partial consensus between neighbouring nodes through the general equality constraints. It becomes a full consensus problem when the set of edge constraints are reduced to $\{\boldsymbol{x}_i = \boldsymbol{x}_j | (i,j) \in \mathcal{E}\}$. In the literature, a majority of research has focused on the full consensus problem, such as the aforementioned work [6–9, 11–13]. In recent years, new applications over WSNs have emerged which only impose partial consensus among neighbouring nodes as represented by (1) (see [1, 21–24]).

Theoretical convergence analysis of synchronous and asynchronous PDMM is provided in [20] and [25] for decomposable convex functions. [20] makes use of variational inequality (VI) to conduct the analysis while [25] relies on the monotonic operator theory. The algorithm has been applied successfully for solving a number of practical problems, which include distributed dictionary learning [26], distributed support vector machine (SVM) [27], distributed speech enhancement over a wireless microphone network (see [24, 29]), and distributed image fusion [22].

Our recent work [23] has considered simplifying the computation of synchronous PDMM for solving a subclass of the optimization (1) with the set of equality constraints $\{\boldsymbol{B}_{i|j}\boldsymbol{x}_i = \boldsymbol{B}_{j|i}\boldsymbol{x}_j | (i,j) \in \mathcal{E}\}$. One simplification made in [23] is to approximate each individual function f_i in (1) by a quadratic function in the updating procedure of PDMM. The approximation is realized by making use of the gradient information of the objective function at each iteration. The motivation behind this is that quadratic functions are computationally cheaper to handle than other functions (see [17] for simplifying ADMM using gradient information). Convergence analysis has been provided in [23] for approximating strongly convex functions with Lipschitz continuous gradient.

In this paper, we provide a new convergence analysis for the gradient-based PDMM (GPDMM) considered in [23]. We show that if each objective function f_i in (1) has Lipschitz continuous gradient, synchronous GPDMM converges to an optimal solution at the sublinear rate $\mathcal{O}(1/K)$ (where K denotes the iteration index). The objective functions do not have to be strongly convex for the al-

gorithm to work in comparison to the analysis in [23]. The new analysis makes use of the analysis approach in [30] developed for the fast iterative-shrinkage thresholding algorithm (FISTA). Experimental results on a problem of ridge regularized logistic regression (RRLR) show that synchronous GPDMM is computationally much cheaper than synchronous PDMM per iteration. As a result, synchronous GPDMM takes much less computational time to obtain a satisfactory solution to the RRLR problem than synchronous PDMM.

2. PROBLEM DEFINITION

We denote an undirected graph as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, m\}$ represents the set of nodes and $\mathcal{E} = \{(i,j)|i,j \in \mathcal{V}\}$ represents the set of undirected edges in the graph, respectively. If $(i,j) \in \mathcal{E}$, node i and j can communicate with each other directly along their edge (i,j). We use $\mathcal{N}_i$ to denote the set of all neighbouring nodes of node i, i.e., $\mathcal{N}_i = \{j|(i,j) \in \mathcal{E}\}$.

With the notation $G = (\mathcal{V}, \mathcal{E})$ for a graph, we consider a subclass of the decomposable convex optimization (1), given by

$$\min_{\boldsymbol{x}} \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i) \quad \text{s.t.} \quad \boldsymbol{B}_{i|j}\boldsymbol{x}_i = \boldsymbol{B}_{j|i}\boldsymbol{x}_j \quad \forall (i,j) \in \mathcal{E}, \qquad (2)$$

where $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{|\mathcal{V}|}]^T$, and each convex function $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is continuously differentiable with the Lipschitz continuous gradient $L_i(f_i) > 0$:

$$\|\nabla f_i(\boldsymbol{x}_i) - \nabla f_i(\boldsymbol{y}_i)\| \leq L_i(f_i)\|\boldsymbol{x}_i - \boldsymbol{y}_i\| \quad \forall \boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{R}^{n_i}, \quad (3)$$

where $\|\cdot\|$ denotes the standard Euclidean norm. The vector $\boldsymbol{x}$ is thus of dimension $n_{\boldsymbol{x}} = \sum_{i \in \mathcal{V}} n_i$. For every edge $(i,j) \in \mathcal{E}$, we have $(\boldsymbol{B}_{i|j}, \boldsymbol{B}_{j|i}) \in (\mathbb{R}^{n_{ij} \times n_i}, \mathbb{R}^{n_{ij} \times n_j})$. In general, $\boldsymbol{B}_{i|j}$ and $\boldsymbol{B}_{j|i}$ are two different matrices. The matrix $\boldsymbol{B}_{i|j}$ operates on $\boldsymbol{x}_i$ in the linear constraint of edge $(i,j) \in \mathcal{E}$.

We assume there exists an optimal solution $\boldsymbol{x}^\star$ to the above problem (2). The research goal is to compute or obtain a good approximation of $\boldsymbol{x}^\star$ via local computation and transmission between neighbouring nodes distributedly after a reasonably number of iterations. To achieve the goal, the main challenge is to decide what information should be sent from a node to its neighbours per iteration and how to make use of the received information at each node for local computation.

3. GRADIENT BASED PRIMAL-DUAL METHOD OF MULTIPLIERS (GPDMM)

We first present the updating procedure of synchronous PDMM for solving (2). To do so, we introduce a set of auxiliary variables for the algorithm to work. Let $\boldsymbol{\lambda}_{i|j}$ and $\boldsymbol{\lambda}_{j|i}$ be two auxiliary variables for every edge constraint $\boldsymbol{B}_{i|j}\boldsymbol{x}_i = \boldsymbol{B}_{j|i}\boldsymbol{x}_j$. The variable $\boldsymbol{\lambda}_{i|j}$ is owned by and updated at node i and is related to neighbour j. We denote by $\boldsymbol{\lambda}_i$ the concatenation of all $\boldsymbol{\lambda}_{i|j}$, $j \in \mathcal{N}_i$. Therefore each node i carries two variables $\boldsymbol{x}_i$ and $\boldsymbol{\lambda}_i$. Similarly to $\boldsymbol{x}$, we let $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T, \ldots, \boldsymbol{\lambda}_{|\mathcal{V}|}^T]^T$.

Synchronous PDMM updates $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ simultaneously per iteration by performing node-oriented computation. At iteration k, each i computes a new estimate $\boldsymbol{x}_i^{k+1}$ by locally solving a small-size optimization problem. In doing so, the neighbouring estimates $\{\boldsymbol{x}_j^k | j \in \mathcal{N}_i\}$ and $\{\boldsymbol{\lambda}_{j|i}^k | j \in \mathcal{N}_i\}$ from last iteration are used. Once $\boldsymbol{x}_i^{k+1}$ is obtained, the estimates $\{\boldsymbol{\lambda}_{i|j}^{k+1} | j \in \mathcal{N}_i\}$ can then be computed. The updating expressions for $\boldsymbol{x}_i^{k+1}$ and $\{\boldsymbol{\lambda}_{i|j}^{k+1} | j \in \mathcal{N}_i\}$ are given by [23]

$$\boldsymbol{x}_i^{k+1} = \arg\min_{\boldsymbol{x}_i} \Big[f_i(\boldsymbol{x}_i) + \sum_{j \in \mathcal{N}_i} \boldsymbol{\lambda}_{j|i}^{k,T} \boldsymbol{B}_{i|j}\boldsymbol{x}_i + \frac{1}{2}\|\boldsymbol{B}_{i|j}\boldsymbol{x}_i - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^k\|_{\boldsymbol{P}_{ij}}^2 \Big] \quad i \in \mathcal{V} \qquad (4)$$

$$\boldsymbol{\lambda}_{i|j}^{k+1} = \boldsymbol{P}_{ij}(\boldsymbol{B}_{j|i}\boldsymbol{x}_j^k - \boldsymbol{B}_{i|j}\boldsymbol{x}_i^{k+1}) - \boldsymbol{\lambda}_{j|i}^k \quad i \in \mathcal{V}, j \in \mathcal{N}_i \quad (5)$$

where each $\boldsymbol{P}_{ij}$ is a positive definite matrix (i.e., $\boldsymbol{P}_{ij} \succ 0$), and $\|\cdot\|_{\boldsymbol{P}}$ represents the weighted Euclidean norm by the matrix $\boldsymbol{P}$. We let $\mathcal{P} = \{\boldsymbol{P}_{ij} \succ 0 | (i,j) \in \mathcal{E}\}$, which remains to be specified.

We note that for some objective functions $\{f_i | i \in \mathcal{V}\}$ (e.g., softmax function in logistic regression), it might be expensive to compute the exact solution $\{\boldsymbol{x}_i^{k+1} | i \in \mathcal{V}\}$ in (4). In those situations, GPDMM attempts to simplify the optimization (4) by using the gradient information of the objective function computed at the most recent estimate. To do so, an approximation of each individual function f_i at iteration k is defined as (see [23])

$$f_i^k(\boldsymbol{x}_i) = f_i(\boldsymbol{x}_i^k) + (\boldsymbol{x}_i - \boldsymbol{x}_i^k)^T \nabla f_i(\boldsymbol{x}_i^k) + \frac{L_i}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_i^k\|^2 \quad i \in \mathcal{V}, \quad (6)$$

where $L_i > 0$. Replacing $f_i(\boldsymbol{x}_i)$ with the approximation $f_i^k(\boldsymbol{x}_i)$ in (4)-(5) produces the updating expressions of synchronous GPDMM

$$\boldsymbol{x}_i^{k+1} = \arg\min_{\boldsymbol{x}_i} \Big[f_i^k(\boldsymbol{x}_i) + \sum_{j \in \mathcal{N}_i} \boldsymbol{\lambda}_{j|i}^{k,T} \boldsymbol{B}_{i|j}\boldsymbol{x}_i + \frac{1}{2}\|\boldsymbol{B}_{i|j}\boldsymbol{x}_i - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^k\|_{\boldsymbol{P}_{ij}}^2 \Big] \quad i \in \mathcal{V} \qquad (7)$$

$$\boldsymbol{\lambda}_{i|j}^{k+1} = \boldsymbol{P}_{ij}(\boldsymbol{B}_{j|i}\boldsymbol{x}_j^k - \boldsymbol{B}_{i|j}\boldsymbol{x}_i^{k+1}) - \boldsymbol{\lambda}_{j|i}^k \quad i \in \mathcal{V}, j \in \mathcal{N}_i. \quad (8)$$

With (6)-(8), the optimality condition for each $\boldsymbol{x}_i^{k+1}$ can be easily derived as

$$\nabla f_i(\boldsymbol{x}_i^k) + L_i(\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k) = \sum_{j \in \mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^{k+1} \quad i \in \mathcal{V}, \qquad (9)$$

which will be used for the convergence analysis in next section.

Synchronous GPDMM converges to an optimal solution $\lim_{k\to\infty}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = (\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ for the problem (2) if and only if $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ satisfies the following optimality conditions [20, 23]

$$\nabla f_i(\boldsymbol{x}_i^\star) = \sum_{j \in \mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \qquad i \in \mathcal{V} \qquad (10)$$

$$\boldsymbol{\lambda}_{i|j}^\star = -\boldsymbol{\lambda}_{j|i}^\star \qquad (i,j) \in \mathcal{E} \qquad (11)$$

$$\boldsymbol{B}_{i|j}\boldsymbol{x}_i^\star = \boldsymbol{B}_{j|i}\boldsymbol{x}_j^\star \qquad (i,j) \in \mathcal{E}. \qquad (12)$$

We will show in next section that if the set of parameters $\{L_i | i \in \mathcal{V}\}$ are properly chosen in (6), synchronous GPDMM would converge to an optimal solution.

4. CONVERGENCE ANALYSIS

In this section, we present the new convergence analysis for synchronous GPDMM in comparison to [23]. Inspired by the analysis for FISTA [30], we first construct a special inequality for each $\boldsymbol{x}_i^{k+1}$ in (7) and then exploit it to analyze synchronous GPDMM.

4.1. Constructing an inequality

Before formally presenting the inequality, we first introduce a standard inequality for each f_i with the Lipschitz continuous gradient $L_i(f_i)$ in (3):

Lemma 1 (Prop. A24 in [31])**.** *Let each f_i in (2) be a continuously differentiable function with the Lipschitz continuous gradient $L_i(f_i)$. Then for any $L_i \geq L_i(f_i)$,*

$$f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) + (\boldsymbol{x}-\boldsymbol{y})^T \nabla f_i(\boldsymbol{y}) + \frac{L_i}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

With Lemma 1, we are now ready to derive the inequality for each $\boldsymbol{x}_i^{k+1}$ in (7):

Lemma 2. *Let $L_i \geq L_i(f_i)$ in the approximation function (6). Then for any $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$,*

$$\begin{aligned} f_i(\boldsymbol{x}_i) - f_i(\boldsymbol{x}_i^{k+1}) \geq & \frac{L_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k\|^2 - L_i(\boldsymbol{x}_i-\boldsymbol{x}_i^k)^T(\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k) \\ & + (\boldsymbol{x}_i-\boldsymbol{x}_i^{k+1})^T \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^{k+1}. \end{aligned} \tag{13}$$

Proof. From (6) and Lemma 1, we have

$$\begin{aligned} & f_i(\boldsymbol{x}_i) - f_i(\boldsymbol{x}_i^{k+1}) \\ \geq & f_i(\boldsymbol{x}_i) - f_i^k(\boldsymbol{x}_i^{k+1}) \\ \overset{(a)}{\geq} & f_i(\boldsymbol{x}_i^k) + (\boldsymbol{x}_i - \boldsymbol{x}_i^k)^T \nabla f_i(\boldsymbol{x}_i^k) - f_i^k(\boldsymbol{x}_i^{k+1}) \\ = & f_i(\boldsymbol{x}_i^k) + (\boldsymbol{x}_i - \boldsymbol{x}_i^k)^T \nabla f_i(\boldsymbol{x}_i^k) \\ & - \left[f_i(\boldsymbol{x}_i^k) + (\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k)^T \nabla f_i(\boldsymbol{x}_i^k) + \frac{L_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k\|^2 \right] \\ = & (\boldsymbol{x}_i-\boldsymbol{x}_i^{k+1})^T \nabla f_i(\boldsymbol{x}_i^k) - \frac{L_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k\|^2 \\ \overset{(b)}{=} & (\boldsymbol{x}_i-\boldsymbol{x}_i^{k+1})^T \left(\sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^{k+1} - L_i(\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k) \right) \\ & - \frac{L_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^k\|^2, \end{aligned}$$

where step (a) uses the property that f_i is a convex function and step (b) uses (9). By using algebra, the above expression can be further simplified as (13). The proof is complete. □

4.2. Convergence properties

In this subsection, we derive the convergence properties of synchronous GPDMM based on Lemma 2. The derivation procedure is similar to our early work [20] for analyzing synchronous PDMM.

Suppose $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ is an optimal solution satisfying (10)-(12). We first derive an upper and lower bound for a quantity $\sum_{i\in\mathcal{V}}\left[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i^\star) - \boldsymbol{x}_i^{k+1,T} \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \right]$ in a lemma below.

Lemma 3. *Let $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ be an optimal solution satisfying (10)-(12). The estimate $(\boldsymbol{x}^{k+1}, \boldsymbol{\lambda}^{k+1})$ is obtained by performing (6)-(8) under the condition that $L_i \geq L_i(f_i)$, $i \in \mathcal{V}$. Then there is*

$$\begin{aligned} 0 \leq & 2\sum_{i\in\mathcal{V}} \left[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i^\star) - \boldsymbol{x}_i^{k+1,T} \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \right] \\ \leq & \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \Big[\|\boldsymbol{P}_{ij}^{-1/2}(\boldsymbol{\lambda}_{i|j}^* + \boldsymbol{\lambda}_{j|i}^k) + \boldsymbol{P}_{ij}^{1/2}(\boldsymbol{B}_{i|j}\boldsymbol{x}_i^* - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^k)\|^2 \\ & - \|\boldsymbol{P}_{ij}^{-1/2}(\boldsymbol{\lambda}_{i|j}^* + \boldsymbol{\lambda}_{j|i}^{k+1}) + \boldsymbol{P}_{ij}^{1/2}(\boldsymbol{B}_{i|j}\boldsymbol{x}_i^* - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^{k+1})\|^2 \\ & - \|\boldsymbol{P}_{ij}^{-1/2}(\boldsymbol{\lambda}_{i|j}^{k+1} + \boldsymbol{\lambda}_{j|i}^k) + \boldsymbol{P}_{ij}^{1/2}(\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{k+1} - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^k)\|^2 \Big] \\ & + \sum_{i\in\mathcal{V}} L_i\|\boldsymbol{x}_i^k - \boldsymbol{x}_i^\star\|^2 - \sum_{i\in\mathcal{V}} L_i\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^\star\|^2 \end{aligned} \tag{14}$$

where the equality for the lower bound holds if and only if $(\boldsymbol{x}^{k+1}, \boldsymbol{\lambda}^{k+1})$ satisfies

$$\nabla f_i(\boldsymbol{x}_i^{k+1}) = \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{j|i}^\star \qquad \forall i \in \mathcal{V}. \tag{15}$$

Proof. The complete proof for the upper bound takes a lot of space. We therefore only explain the basic idea of the proof. It includes two main steps. Firstly, we invoke Lemma 2 with $\boldsymbol{x}_i = \boldsymbol{x}_i^\star$, and then summarize the inequality over all $i \in \mathcal{V}$. Secondly, we reformulate the obtained inequality to produce (14) by using (8), (11)-(12) and the identity

$$\begin{aligned} & (\boldsymbol{y}_1 - \boldsymbol{y}_2)^T(\boldsymbol{y}_3 - \boldsymbol{y}_4) \\ \equiv & \frac{1}{2}(\|\boldsymbol{y}_1 + \boldsymbol{y}_3\|^2 - \|\boldsymbol{y}_2 + \boldsymbol{y}_4\|^2 - \|\boldsymbol{y}_2 + \boldsymbol{y}_3\|^2 + \|\boldsymbol{y}_2 + \boldsymbol{y}_4\|^2). \end{aligned}$$

See a similar derivation for the proof for Lemma 8 in [20].

Next we prove the lower bound of (14).

$$\begin{aligned} & 2\sum_{i\in\mathcal{V}} \left[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i^\star) - \boldsymbol{x}_i^{k+1,T} \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \right] \\ \overset{(a)}{\geq} & 2\sum_{i\in\mathcal{V}} \Big[- f_i^*\Big(\sum_{i\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \Big) - f_i(\boldsymbol{x}_i^\star) \Big] \\ \overset{(b)}{=} & 0, \end{aligned}$$

where $f_i^*(\cdot)$ is the conjugate function of $f_i(\cdot)$, step (a) uses Fenchel's inequality (see [32]), and step (b) uses (11)-(12) and the definition of the conjugate functions $\{f_i^* | i \in \mathcal{V}\}$. The equality in step (a) holds when (15) is satisfied. The proof is complete.

□

Next we show that the estimates $(\boldsymbol{x}^{k+1}, \boldsymbol{\lambda}^{k+1})$ are always bounded using the results of Lemma 3.

Lemma 4. *Every pair of estimates $(\hat{\boldsymbol{x}}_i{}^{k+1}, \hat{\boldsymbol{\lambda}}_{i|j}^{k+1})$, $i \in \mathcal{V}$, $j \in \mathcal{N}_i$, $k \geq 0$, in Lemma 3 is upper bounded by a constant M under a squared error criterion:*

$$\left\| \boldsymbol{P}_{ij}^{-\frac{1}{2}}(\boldsymbol{\lambda}_{i|j}^\star + \boldsymbol{\lambda}_{j|i}^{k+1}) + \boldsymbol{P}_{ij}^{\frac{1}{2}}(\boldsymbol{B}_{i|j}\boldsymbol{x}_i^\star - \boldsymbol{B}_{j|i}\boldsymbol{x}_j^{k+1}) \right\|^2 \leq M. \tag{16}$$

Proof. One can first prove (16) for $k = 0$ by using (14). The inequality (16) for $k > 0$ can then be proved recursively. □

Upon obtaining the results in Lemma 3 and 4, we are ready to present the convergence rate of synchronous GPDMM.

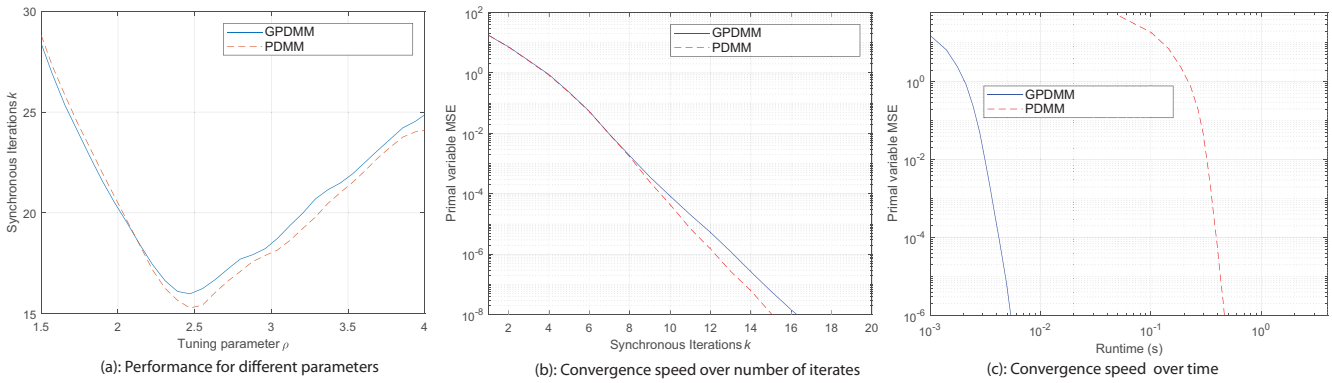

Fig. 1. Performance comparison of synchronous GPDMM and PDMM for the ridge regularized logistic regression. Subplot (b) and (c) were obtained by using the optimal parameter $\rho^\star = 2.4$.

Theorem 1. *Let $(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$, $k = 1, \ldots, K$, be obtained by performing (6)-(8) under the condition that $L_i \geq L_i(f_i)$, $i \in \mathcal{V}$. The average estimate $(\bar{\boldsymbol{x}}^K, \bar{\boldsymbol{\lambda}}^K) = (\frac{1}{K}\sum_{k=1}^K \boldsymbol{x}^k, \frac{1}{K}\sum_{k=1}^K \boldsymbol{\lambda}^k)$ satisfies*

$$0 \leq \sum_{i\in\mathcal{V}} \Big[f_i(\bar{\boldsymbol{x}}_i^K) - f_i(\boldsymbol{x}_i^\star) - \bar{\boldsymbol{x}}_i^{K,T} \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^T \boldsymbol{\lambda}_{i|j}^\star \Big] \leq \mathcal{O}\Big(\frac{1}{K}\Big) \quad (17)$$

$$\lim_{K\to\infty} \boldsymbol{B}_{i|j}\bar{\boldsymbol{x}}_i^K = \boldsymbol{B}_{j|i}\bar{\boldsymbol{x}}_j^K \quad \forall (i,j) \in \mathcal{E} \quad (18)$$

$$\lim_{K\to\infty} \bar{\boldsymbol{\lambda}}_{i|j}^K + \bar{\boldsymbol{\lambda}}_{j|i}^K = \boldsymbol{0} \quad \forall (i,j) \in \mathcal{E}. \quad (19)$$

Proof. The proof is similar to that for Theorem 2 in [20]. □

The conditions $\{L_i \geq L_i(f_i) | i \in \mathcal{V}\}$ in Theorem 1 ensure that synchronous GPDMM possesses the same convergence rate as synchronous PDMM (see [20]). There is no additional requirement on the matrix set $\mathcal{P}$ for GPDMM to work.

5. EXPERIMENT

In this section we consider solving the problem of ridge regularized logistic regression (RRLR) over a chain graph of 5 nodes. The problem function at each node i is

$$g_i(\boldsymbol{x}_i) = \frac{1}{10}\sum_{p=1}^{10} \log[1 + \exp(-c_{ip}\boldsymbol{d}_{ip}^T\boldsymbol{x}_i)] + \|\boldsymbol{x}_i\|_2^2, \quad (20)$$

where each node i holds 10 training points consisting of feature vector $\boldsymbol{d}_{ip} \in \mathbb{R}^{10}$ and binary label c_{ip} for $p = 1, ..., 10$. The objective is to perform distributed data training in the graph so that after convergence all the nodes reach a common consensus of the optimal solution, i.e., $\lim_{k\to\infty} \boldsymbol{x}_i^k = \boldsymbol{x}^\star$ for all i.

We evaluated both synchronous GPDMM and PDMM using Matlab code on a Windows computer. In the implementation of synchronous PDMM, the L-BFGS algorithm [33] was used to solve local subproblems involving the functions $\{g_i(\boldsymbol{x}_i) | i \in \mathcal{V}\}$. The parameter L_i in (6) for GPDMM was set as $L_i = 2$ for all i. For simplicity, we set all the matrices in $\mathcal{P}$ to be a constant scalar parameter ρ for both methods. That is $\mathcal{P} = \{\boldsymbol{P}_{ij} = \rho | (i,j) \in \mathcal{E}\}$. At each iteration, the mean squared error (MSE) across all nodes in the graph

$$\text{MSE}^k = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \|\boldsymbol{x}_i^k - \boldsymbol{x}^\star\|^2,$$

was measured, where the global optimal solution $\boldsymbol{x}^\star$ was computed beforehand.

Two simulations were conducted for performance comparison of the two methods. In the first simulation, a range of ρ values between $[1.5, 4]$ with intervals of 0.08 were tested. For each ρ value, we counted the number of iterations needed before each method reaches an MSE of 10^{-8}. The two curves in Fig. 1:(a) were obtained by averaging the results of 20 instances. It is seen from Subplot (a) that GPDMM has a slight performance degradation compared to PDMM when $\rho > 2.2$. When $\rho^\star = 2.4$, both methods exhibit the fastest converge speed.

In the second simulation, we studied the convergence speed of the two methods versus the number of iterations and absolute runtime (in units of second) by using $\rho^\star = 2.4$, respectively. The results are displayed in Fig. 1:(b) and (c), respectively. The curves in both subplots are obtained by averaging the results of 500 instances. It is clear that GPDMM is much faster than PDMM in terms of runtime. This suggests that the gradient based distributed computation is much cheaper than tackling the original function $g_i(\boldsymbol{x}_i)$ in (20) for the RRLR problem.

Remark 1. *More experiments have been provided in [23] for synchronous GPDMM over random graphic models rather than simple chain graphs. The results in [23] also confirm that synchronous GPDMM is computationally more efficient than synchronous PDMM.*

6. CONCLUSION

In this paper we have provided a new convergence analysis for synchronous GPDMM. The new analysis only requires the objective functions to have Lipschitz continuous gradient while the analysis in [23] additionally requires the functions to be strongly convex. A sufficient condition on how to setup the parameters of synchronous GPDMM has been provided. Experimental results for the RRLR problem demonstrate that synchronous GPDMM is computationally much cheaper than synchronous PDMM. This suggests that if PDMM can not obtain a closed form solution in its updating expressions for the considered objective functions, GPDMM should then be considered for cheaper local computations.

7. REFERENCES

[1] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip Algorithms for Distributed Signal Processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[2] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, Springer, 2015.

[3] C. C. Moallemi and B. Van Roy, "Convergence of Min-Sum Message Passing for Quadratic Optimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2413–2423, 2009.

[4] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1(1-2), pp. 1–305, 2008.

[5] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," *Morgan Kaufman Publishers*, 1988.

[6] J. Duchi, A. Agarwal, and M. J. Wainwright, "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling," in *IEEE Trans. Automatic Control*, 2012, vol. 57, pp. 592–606.

[7] A. Nedić and A. Ozdaglar, "Distributed Subgradient Methods for Multi-agent Optimization," *IEEE Transactions on Automatic Control*, 2008.

[8] J. Chen and A.H. Sayed, "Diffusion Adaptation Strategies for Distributed Optimization and Learning Over Networks," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, pp. 944–966, 2014.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *In Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] D. Hajinezhad and M. Hong, "Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015.

[12] P. D. Lorenzo and G. Scutari, "NEXT: In-Network Nonconvex Optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[13] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The Proximal Primal-Dual Algorithm for Fast Distributed Nonconvex Optimization and Learning Over Networks," in *International Conference on Machine Learning (ICML)*, 2017.

[14] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the Linear Convergence of the ADMM in Decentralized Consensus Optimization," *IEEE Trans. Signal Processing*, vol. 7, pp. 1750–1761, 2014.

[15] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 162, pp. 165–199, 2017.

[16] T.-H. Chang, M. Hong, and X. Wang, "Multi-Agent Distributed Optimization via Inexact Consensus ADMM," *IEEE Trans. Signal Processing*, vol. 63, pp. 482–497, 2015.

[17] Q. Ling, W. Shi, G. Wu, and A. Ribeiron, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Processing*, vol. 63, pp. 4051–4064, 2015.

[18] N. Komodakis and J.-C. Pesquet, "Playing with Duality: An Overview of Recent Primal-Dual Approaches for Solving Large-Scale Optimization problems," *I IEEE Signal Processing Magazine*, vol. 32, pp. 31–54, 2014.

[19] J.-C. Pesquet and A. Repetti, "A Class of Randomized Primal-Dual Algorithms for Distributed Optimization," *J. Nonlinear Convex Anal.*, vol. 16, pp. 2353–2490, 2015.

[20] G. Zhang and R. Heusdens, "Distributed Optimization using the Primal-Dual Method of Multipliers," IEEE Trans. Signal and Information Processing over Networks, 2017.

[21] G. Zhang, W. B. Kleijn, and R. Heusdens, "On Relationship between Primal-Dual Method of Multipliers and Kalman Filter," arXiv:1708.06881 [math.OC], 2017.

[22] M. O. Connor, W. B. Kleijn, and T. Abhayapala, "Distributed TV-L1 Image Fusion Using PDMM," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017, pp. 3326–3330.

[23] M. O. Connor and G. Zhang and W. B. Kleijn and T. Abhayapala, "Function Splitting and Quadratic Approximation of the Primal-Dual Method of Multipliers for Distributed Optimization over Graphs," accepted by IEEE. Trans. Signal and Information Processing over Networks, 2018.

[24] M. O. Connor, W. B. Kleijn, and T. Abhayapala, "Distributed sparse MVDR beamforming using the bi-alternating direction method of multipliers," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 106–110.

[25] T. Sherson and R. Heusdens, W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," arXiv:1706.02654 [math.OC], 2017.

[26] H. M. Zhang, "Distributed Convex Optimization: A Study on the Primal-Dual Method of Multipliers," M.S. thesis, Delft University of Technology, 2015.

[27] G. Zhang and R. Heusdens, "On Simplifying the Primal-Dual Method of Multipliers," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2016, pp. 4826–4830.

[28] V. M. Tavakoli, J. R. Jensen, R. Heusdens, J. Benesty, and M. G. Christensen, "Distributed max-SINR speech enhancement with ad hoc microphone arrays," in *ICASSP*, 2017, pp. 151–155.

[29] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[30] D. P. Bertsekas, *Nonlinear Programming*, Belmont, MA: Athena Scientific, 2nd edition, 1999.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[32] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.