GENERALIZED DISTRIBUTED DUAL COORDINATE ASCENT IN A TREE NETWORK FOR MACHINE LEARNING

Myung Cho[1]*, Lifeng Lai*[2]*, and Weiyu Xu*[3]

[1] Dept. of ECE, North Carolina State University, Raleigh, NC, 27606
[2] Dept. of ECE, University of California, Davis, CA, 95616
[3] Dept. of ECE, University of Iowa, Iowa City, IA, 52242

ABSTRACT

With explosion of data size and limited storage space at a single location, data are often distributed at different locations. We thus face the challenge of performing large-scale machine learning from these distributed data through communication networks. In this paper, we generalize the distributed dual coordinate ascent in a star network to a general tree structured network, and provide the convergence rate analysis of the general distributed dual coordinate ascent. In numerical experiments, we demonstrate that the performance of the distributed dual coordinate ascent in a tree network can outperform that of the distributed dual coordinate ascent in a star network when a network has a lot of communication delays between the center node and its direct child nodes.

Index Terms— gradient descent, machine learning, distributed system, dual coordinate ascent, big data

1. INTRODUCTION

In modern society, the amount of data that we can access and learn information from is skyrocketing due to the abundance of sensors. This propels our society into an era of *big data* [1]. However, data are very often collected and stored at different locations, due to the constraints of limited storage volumes and network communication bandwidths. This necessitates performing machine learning in a distributed manner.

In order to answer the challenge of distributed data, researchers have studied various optimization methods such as synchronous Stochastic Gradient Decent (SGD) [2, 3], synchronous Stochastic Dual Coordinate Ascent (SDCA) [4–6], asynchronous SGD [7, 8], and asynchronous SDCA [9, 10]. Even though the convergence of SGD does not depend on the size of data, it is reported in [11] that SDCA can outperform SGD when we need relatively high solution accuracy. Furthermore, asynchronous updating scheme can suffer from the conflicts between intermediate results.

Motivated by these facts, the authors in [4–6] considered a synchronous distributed dual coordinate ascent for solving regularized loss minimization problems in a star network. In this star network, data are distributed over a few local workers, which can individually communicate with a central station. In [4–6], the authors derived the convergence rate of the distributed dual coordinate ascent with respect to the number of iterations. The proposed distributed optimization framework in [5, 6] is free of tuning parameters or learning rates, compared with SGD-based methods. Moreover, the duality gap in [5, 6] readily provides a fair stopping criterion and efficient accuracy certificates.

However, practical communication networks are not always organized in a star network, but sometimes in very different network topologies. It is unclear how to design and analyze dual coordinate ascent algorithms for a network with general topologies. In addition, it is unknown how network communication delays (not merely the number of communication rounds) will affect the design and convergence rate of distributed dual coordinate ascent algorithms [4–6]. We remark that, in [12], the authors considered communication delays and provided the convergence bound in terms of time for consensus based distributed optimization.

In this paper, we generalize the previous research [4–6] on the distributed dual coordinate ascent in a star network. Especially, we consider the design of the distributed dual coordinate ascent algorithms for regularized loss minimization, in a general *tree structured* network. And then, we provide the convergence rate analysis of the general distributed dual coordinate ascent in the considered tree network. In the numerical experiments, we demonstrate that when the communication delays between the center node and its direct child nodes are large, the generalized distributed dual coordinate ascent in a tree network can have better convergence speed than that in a star network.

Notations: We use $[k]$ to denote the index set of the coordinates in the k-th coordinate block. For an index set Q, $\overline{Q}$ and $|Q|$ represent the complement and the cardinality of Q respectively. We use bold letters for vectors and matrices. Using an index set as a subscript of a vector (matrix) refers to the partial vector (partial matrix) over the index set (with columns over the index set). The superscript (t) is used to denote the t-th iteration. The superscript $\star$ is reserved for the optimal solution.

2. PROBLEM SETUP

We have the following regularized loss minimization problem for machine learning applications [4, 5, 9, 10, 13]:

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\ P(\boldsymbol{w}) \triangleq \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\ell_i(\boldsymbol{w}^T\boldsymbol{x}_i), \qquad (2.1)$$

The work of W. Xu was supported in part by Simons Foundation 318608 and in part by National Science Foundation (NSF) DMS-1418737. And the work of L. Lai was supported by NSF under grants CCF-1717943, ECCS-1711468 and CNS-1824553.

Algorithm 1: Distributed Dual Coordinate Ascent [5]

Input: $T \geq 1$
Output: $\boldsymbol{w}, \boldsymbol{\alpha}$
Data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ distributed over K local workers
Initialization: $\boldsymbol{\alpha}_{[k]}^{(0)} \leftarrow 0$ for all local workers, and $\boldsymbol{w}^{(0)} \leftarrow 0$
for $t = 1$ **to** T **do**
  for *all local workers* $k = 1, 2, ..., K$ *in parallel* **do**
    $(\Delta\boldsymbol{\alpha}_{[k]}, \Delta\boldsymbol{w}_k) \leftarrow$ LocalDualMethod$(\boldsymbol{\alpha}_{[k]}^{(t-1)}, \boldsymbol{w}^{(t-1)})$
    $\boldsymbol{\alpha}_{[k]}^{(t)} \leftarrow \boldsymbol{\alpha}_{[k]}^{(t-1)} + \frac{1}{K}\Delta\boldsymbol{\alpha}_{[k]}$
  end
  $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \frac{1}{K}\sum_{k=1}^K \Delta\boldsymbol{w}_k$
end

where $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, ..., m$, are dataset, $\ell_i(\cdot)$, $i = 1, ..., m$, are loss functions, and λ is the regularization parameter. Throughout the paper, we assume that $\|\boldsymbol{x}_i\| \leq 1$, $\forall i$. Depending on the loss functions, one can consider (2.1) as various machine learning problems ranging from regression to classification. For example, if the loss function is the hinge loss function, the optimization problem with labeled dataset $\{(\boldsymbol{x}_i, y_i)\}$, $i = 1, ..., m$, where $y_i \in \mathbb{R}$ is label information, becomes the Support Vector Machine (SVM).

Using the conjugate function, i.e., $\ell_i(a) = \sup_b ab - \ell_i^*(b)$, where $a, b \in \mathbb{R}$ and $\ell_i(\cdot)$ is convex, the dual problem of (2.1) is stated as

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\text{maximize}}\ D(\boldsymbol{\alpha}) \triangleq -\frac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{\alpha}\|^2 - \frac{1}{m}\sum_{i=1}^m \ell_i^*(-\alpha_i), \tag{2.2}$$

where α_i is the i-th element of the dual vector $\boldsymbol{\alpha}$, and the data matrix $\boldsymbol{A} \in \mathbb{R}^{d\times m}$ has the normalized training data $\frac{1}{\lambda m}\boldsymbol{x}_i$ in its i-th column, i.e., $\boldsymbol{A}_i = \frac{1}{\lambda m}\boldsymbol{x}_i$. Due to the primal-dual relationship as $\boldsymbol{w}(\boldsymbol{\alpha}) \triangleq \boldsymbol{A}\boldsymbol{\alpha}$, we have the duality gap as $P(\boldsymbol{w}(\boldsymbol{\alpha})) - D(\boldsymbol{\alpha})$.

We consider a distributed dual coordinate ascent for the regularized loss minimization problem over distributed data in a network of computers. Let us review the previous research on the distributed dual coordinate ascent in a star network in the following section.

3. REVIEW OF DISTRIBUTED DUAL COORDINATE ASCENT IN A STAR NETWORK

The authors in [4–6] consider a star network as shown in Fig. 1 and assume that each local worker has disjoint parts of dataset. Specifically, the k-th local worker has training data $\{(\boldsymbol{x}_i, y_i)\}$, $i \in [k]$, where $[k]$ is the index set for the training data of the k-th local worker. Hence, if the star network has K local workers, $|\cup_{k=1}^K [k]| = m$. In [5], the authors introduced Algorithm 1 for the distributed dual coordinate ascent, so-called CoCoA [5]. Later, the authors proposed the updated CoCoA, so-called CoCoA+ in [6]. In Algorithm 1, LocalDualMethod(·) represents any dual method to solve (2.2). The Stochastic Dual Coordinate Ascent (SDCA), denoted by LocalSDCA(·), is a possible candidate for LocalDualMethod(·) [5, 6]. The convergence rate of Algorithm 1 is given as follows [5].

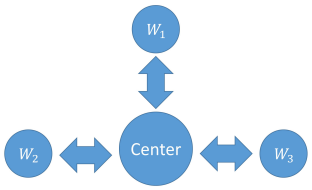

Fig. 1. A star network, where W_i, $i = 1, 2, 3$, are local workers.

Theorem 3.1 ([5, Theorem 2]). *Assume that Algorithm 1 is run for T outer iterations of K local computers, with the procedure LocalSDCA(·) having local geometric improvement Θ. Further, assume the loss functions $\ell_i(\cdot)$ are $1/\gamma$-smooth. Then, the following geometric convergence rate holds for the global (dual) objective:*

$$E[D(\boldsymbol{\alpha}^*) - D(\boldsymbol{\alpha}^{(T)})] \leq \left(1 - (1-\Theta)\frac{1}{K}\frac{\lambda m\gamma}{\rho + \lambda m\gamma}\right)^T \left(D(\boldsymbol{\alpha}^*) - D(\boldsymbol{\alpha}^{(0)})\right), \tag{3.1}$$

where ρ is any real number satisfying

$$\rho \geq \rho_{min} \triangleq \underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\textit{maximize}}\ \lambda^2 m^2 \frac{\sum_{k=1}^K \|\boldsymbol{A}_{[k]}\boldsymbol{\alpha}_{[k]}\|^2 - \|\boldsymbol{A}\boldsymbol{\alpha}\|^2}{\|\boldsymbol{\alpha}\|^2} \geq 0.$$

For LocalSDCA(·), Θ can be the following value [5]: $\Theta = (1 - \frac{s}{\tilde{m}})^H$, where $\tilde{m} \triangleq \max_{k=1,...,K} m_k$ is the size of the largest block of coordinates among K local workers, H is the number of local (or inner) iterations in LocalSDCA(·), and $s \in [0, 1]$ is a step size which determines how far the next solution will be from the current solution at each iteration.

Since a star network that the previous research [4–6] considered is a simple network model, and a network of computers can have various topologies, we study the distributed dual coordinate ascent in a generalized network, specifically, a tree structured network model.

4. GENERALIZED DISTRIBUTED DUAL COORDINATE ASCENT IN A TREE NETWORK

Earlier works [4–6] provide the convergence analysis of the distributed dual coordinate ascent in a star network as shown in Fig. 1. However, the communication network connecting different local workers is not necessarily a simple star network, but instead can be an arbitrary undirected connected graph. The design and analysis of the distributed dual coordinate ascent in a general topology communication network is not well understood as mentioned in [4]. One may argue that, in a network, we can always form a virtual star network by connecting local workers to a central station through the relays of other computers. However, the communication delay from one particular local worker to the central station can be very large (long relays), significantly slowing down the convergence of the distributed learning algorithm. Thus, it is necessary to spend more computational resources on performing distributed optimization among local workers close to each other first, before communicating intermediate computational results to a central station.

Motivated by these network constraints, in this section, we investigate the design and analysis of a recursive distributed dual coordinate ascent algorithm over a general tree

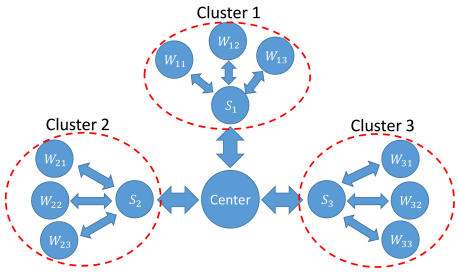

Fig. 2. A tree-structured network, which has two layers. A central station (root node) has three direct child nodes S_1, S_2 and S_3. Each node S_i has three direct local workers W_{ij}, $j = 1, 2, 3$.

Algorithm 2: TreeDualMethod: General Distributed Dual Coordinate Ascent for a General Tree Node Q (not root or leaf)

Input: $T \geq 1$, $\boldsymbol{\alpha}_Q$, $\boldsymbol{w}$

Initialization: $\boldsymbol{\alpha}_{[Q,k]}^{(0)} \leftarrow \boldsymbol{\alpha}_{[Q,k]}$ for all direct child nodes k of node Q , $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{w}$

for $t = 1$ **to** T **do**

  for *all direct child nodes* $k = 1, 2, ..., K$ *of* Q *in parallel* **do**

    $(\triangle\boldsymbol{\alpha}_{[Q,k]}, \triangle\boldsymbol{w}_k) \leftarrow$TreeDualMethod$(\boldsymbol{\alpha}_{[Q,k]}^{(t-1)}, \boldsymbol{w}^{(t-1)})$

    $\boldsymbol{\alpha}_{[Q,k]}^{(t)} \leftarrow \boldsymbol{\alpha}_{[Q,k]}^{(t-1)} + \frac{1}{K}\triangle\boldsymbol{\alpha}_{[Q,k]}$

  end

  $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \frac{1}{K}\sum_{k=1}^{K}\triangle\boldsymbol{w}_k$

end

Output: $\triangle\boldsymbol{\alpha}_Q \triangleq \boldsymbol{\alpha}_Q^{(T)} - \boldsymbol{\alpha}_Q^{(0)}$, and $\triangle\boldsymbol{w}_Q \triangleq \boldsymbol{w}^{(T)} - \boldsymbol{w}^{(0)} = \boldsymbol{A}_Q\triangle\boldsymbol{\alpha}_Q$

Algorithm 3: TreeDualMethod: General Distributed Dual Coordinate Ascent for the Root Node Q

Input: $R \geq 1$

Initialization: $\boldsymbol{\alpha}_{[Q,k]}^{(0)} \leftarrow 0$ for all direct child nodes k of node Q, $\boldsymbol{w}^{(0)} \leftarrow 0$

for $t = 1$ **to** R **do**

  for *all direct child nodes* $k = 1, 2, ..., K$ *in parallel* **do**

    $(\triangle\boldsymbol{\alpha}_{[Q,k]}, \triangle\boldsymbol{w}_k) \leftarrow$TreeDualMethod$(\boldsymbol{\alpha}_{[Q,k]}^{(t-1)}, \boldsymbol{w}^{(t-1)})$

    $\boldsymbol{\alpha}_{[Q,k]}^{(t)} \leftarrow \boldsymbol{\alpha}_{[Q,k]}^{(t-1)} + \frac{1}{K}\triangle\boldsymbol{\alpha}_{[Q,k]}$

  end

  $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + \frac{1}{K}\sum_{k=1}^{K}\triangle\boldsymbol{w}_k$

end

Output: $\boldsymbol{\alpha}^{(R)}$, and $\boldsymbol{w}^{(R)}$

Procedure P. TreeDualMethod: General Distributed Dual Coordinate Ascent for a Leaf Tree Node Q

Input: $H \geq 1$, $\boldsymbol{\alpha}_Q \in \mathbb{R}^{m_Q}$, and $\boldsymbol{w} \in \mathbb{R}^d$ consistent with other coordinate blocks of $\boldsymbol{\alpha}$ s.t. $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{\alpha}$

Data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m_Q}$

Initialization: $\triangle\boldsymbol{\alpha}_Q \leftarrow 0 \in \mathbb{R}^{m_Q}$, and $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{w}$

for $h = 1$ **to** H **do**

  choose $i \in \{1, 2, ..., m_Q\}$ uniformly at random

  find $\triangle\alpha$ maximizing $-\frac{\lambda m}{2}\|\boldsymbol{w}^{(h-1)} + \frac{1}{\lambda m}\triangle\alpha\boldsymbol{x}_i\|^2 - \ell_i^*(-(\alpha_i^{(h-1)} + \triangle\alpha))$

  $\alpha_i^{(h)} \leftarrow \alpha_i^{(h-1)} + \triangle\alpha$

  $(\triangle\boldsymbol{\alpha}_Q)_i \leftarrow (\triangle\boldsymbol{\alpha}_Q)_i + \triangle\alpha$

  $\boldsymbol{w}^{(h)} \leftarrow \boldsymbol{w}^{(h-1)} + \frac{1}{\lambda m}\triangle\alpha\boldsymbol{x}_i$

end

Output: $\triangle\boldsymbol{\alpha}_Q$ and $\triangle\boldsymbol{w}_Q \triangleq \boldsymbol{A}_Q\triangle\boldsymbol{\alpha}_Q$

structured network. We choose to investigate a tree network, because every connected communication network has a spanning tree. In addition, the tree structured network is a generalization of a star network.

We first describe a general tree network, with a 2-layer tree network example illustrated in Fig. 2. In the considered tree network, the root node corresponds to the central station. Any other tree node corresponds to a local worker. Each tree node may have several direct child nodes. Without loss of generality, we assume that only the local workers corresponding to the leaf nodes have access to the distributed data, namely disjoint segmented blocks of the data matrix $\boldsymbol{A}$. We use $[Q, k]$ to denote the set of indices of data stored in the subtree whose root node is the k-th direct child node of Q. If Q is a leaf node, we use m_Q to denote the number of data stored in Q. In a tree network, a node can only communicate with its child nodes or parent nodes.

We are ready to introduce the generalized distributed dual coordinate ascent algorithm (which we call TreeDualMethod) for solving (2.2) dealing with data stored in a general tree structure network. Algorithm 2, Algorithm 3 and Procedure P describe respectively the computational steps of TreeDualMethod for a general tree node (not root or leaf), the root node, and a leaf node. It is noteworthy that in distributed networks, $\triangle\boldsymbol{w}_Q$ (or $\boldsymbol{w}$) is transmitted between nodes, while $\boldsymbol{\alpha}$ (or $\triangle\boldsymbol{\alpha}_Q$) is not. Therefore, when the number of dataset, m, is large, transmitting $\triangle\boldsymbol{w}_Q$ (or $\boldsymbol{w}$) whose dimension is much smaller than m, is beneficial to have communication efficiency. In the next section, we provide the convergence analysis of the generalized distributed dual coordinate ascent in the tree structured network model.

5. CONVERGENCE ANALYSIS OF GENERALIZED DISTRIBUTED DUAL COORDINATE ASCENT IN A TREE NETWORK

We will show that for a tree network, there is a recursive relation between the convergence rate of the algorithm at a tree node Q and the convergence rate at Q's direct child nodes. Suppose that Q has K direct child nodes, and denote the dual variable corresponding to its k-th direct child node by $\boldsymbol{\alpha}_{[Q,k]}$, $1 \leq k \leq K$. We define the local suboptimality gap for Q's k-th direct child node as:

$$\epsilon_{Q,k}(\boldsymbol{\alpha}) \triangleq \underset{\hat{\boldsymbol{\alpha}}_{[Q,k]}}{\text{maximize}}\ D((\boldsymbol{\alpha}_{[Q,1]}, ..., \hat{\boldsymbol{\alpha}}_{[Q,k]}, ..., \boldsymbol{\alpha}_{[Q,K]}, \boldsymbol{\alpha}_{\overline{Q}})) - D((\boldsymbol{\alpha}_{[Q,1]}, ..., \boldsymbol{\alpha}_{[Q,k]}, ..., \boldsymbol{\alpha}_{[Q,K]}, \boldsymbol{\alpha}_{\overline{Q}})), \quad (5.1)$$

Note that the suboptimality gap for the k-th child node is defined when $\boldsymbol{\alpha}_{[Q,i]}$'s ($i \neq k$) and $\boldsymbol{\alpha}_{\overline{Q}}$ are fixed. We further assume that we have the following local geometric improvement for the k-th direct child node of Q.

Assumption 5.1 (Direct child node geometric improvement of TreeDualMethod)**.** *Let us consider a tree node Q. We assume that there exists $\Theta \in [0,1)$ such that for any given $\boldsymbol{\alpha}$, TreeDualMethod for Q's k-th direct child node returns an update $\Delta\boldsymbol{\alpha}_{[Q,k]}$ such that*

$$E[\epsilon_{Q,k}((\boldsymbol{\alpha}_{[Q,1]},...,\boldsymbol{\alpha}_{[Q,k-1]},\boldsymbol{\alpha}_{[Q,k]}+\Delta\boldsymbol{\alpha}_{[Q,k]},...,\boldsymbol{\alpha}_{[Q,K]},\boldsymbol{\alpha}_{\overline{Q}}))] \leq \Theta\cdot\epsilon_{Q,k}(\boldsymbol{\alpha}). \quad (5.2)$$

For a leaf node, TreeDualMethod uses LocalSDCA as in Procedure P. We remark that this geometric improvement condition holds true for the k-th direct child node of Q, i.e., a leaf child node. The following proposition gives a bound on the convergence for a leaf node B even when the input $\boldsymbol{w}$ in Procedure P is also determined by $\boldsymbol{\alpha}_{\overline{Q}}$ and $\boldsymbol{\alpha}_{Q\setminus B}$.

Proposition 5.1 ([5, Proposition 1])**.** *Let us consider a tree node Q whose direct child node B is a leaf node. Assume loss functions $\ell_i(\cdot)$ are $1/\gamma$-smooth. Then for leaf node B, Assumption 5.1 holds with $\Theta = (1-\frac{\lambda m\gamma}{1+\lambda m\gamma}\frac{1}{m_B})^H$, where m_B is the size of data stored at node B.*

Additionally, Theorem 5.2, which is our main result, states that if the geometric improvement condition holds true for direct child nodes of Q, then the geometric improvement also holds true for Q; thus it leads to a recursive calculation of the convergence rate for the tree network.

Theorem 5.2. *Let us consider a tree node Q which has K direct child nodes satisfying the local geometric improvement in Assumption 5.1, with parameters Θ_i, $i=1,...,K$. Suppose that Algorithm 2 (or Algorithm 3) has an input $\boldsymbol{w}$, and Algorithm 2 (or Algorithm 3) is run for T iterations. We further assume that the loss functions $\ell_i(\cdot)$ are $1/\gamma$-smooth.*

Then, for any input $\boldsymbol{w}$ to Algorithm 2 (or Algorithm 3), the following geometric convergence rate holds for Q:

$$E[D(\boldsymbol{\alpha}_Q^*,\boldsymbol{\alpha}_{\overline{Q}})-D(\boldsymbol{\alpha}_Q^{(T)},\boldsymbol{\alpha}_{\overline{Q}})] \quad (5.3)$$
$$\leq \left(1-(1-\Theta)\frac{1}{K}\frac{\lambda m\gamma}{\rho+\lambda m\gamma}\right)^T \left(D(\boldsymbol{\alpha}_Q^*,\boldsymbol{\alpha}_{\overline{Q}})-D(\boldsymbol{\alpha}_Q^{(0)},\boldsymbol{\alpha}_{\overline{Q}})\right),$$

where $\Theta = \max_k \Theta_k$, and ρ is any real number satisfying

$$\rho \geq \rho_{min} \triangleq \underset{\boldsymbol{\alpha}\in\mathbb{R}^{|Q|}}{maximize}\ \lambda^2 m^2 \frac{\sum_{k=1}^K \|\boldsymbol{A}_{[Q,k]}\boldsymbol{\alpha}_{[Q,k]}\|^2 - \|\boldsymbol{A}_Q\boldsymbol{\alpha}_Q\|^2}{\|\boldsymbol{\alpha}_Q\|^2} \geq 0.$$

Because Theorem 5.2 works for any non-leaf tree node, by combining it with Proposition 5.1, we can recursively obtain the convergence rate of the generalized distributed dual coordinate ascent algorithm for the whole tree network. Note that $(1-(1-\Theta)\frac{1}{K}\frac{\lambda m\gamma}{\rho+\lambda m\gamma})^T$ becomes the "Θ" for Q, and (5.3) is seen as the direct child node geometric improvement of TreeDualMethod by the direct parent node of Q.

Theorem 5.2 is different from Theorem 2 of [5] in two aspects. Firstly, Theorem 5.2 works for any tree node in a general tree network, beyond the star network discussed in [5]. Secondly, Theorem 5.2 is true, even when the input $\boldsymbol{w}$ of Algorithm 2 is not only determined by $\boldsymbol{\alpha}_Q$, but also determined by $\boldsymbol{\alpha}_{\overline{Q}}$. To see this, we note that, at the root node, $\boldsymbol{w} = \boldsymbol{A}_Q\boldsymbol{\alpha}_Q + \boldsymbol{A}_{\overline{Q}}\boldsymbol{\alpha}_{\overline{Q}}$, and the root node will pass $\boldsymbol{w}$ to tree node Q by recalling TreeDualMethod$(\cdot)$ for the root node's child nodes. Our proof of Theorem 5.2 addresses this challenge that the input $\boldsymbol{w}$ is also affected by $\boldsymbol{\alpha}_{\overline{Q}}$. Due to the space limitation, we omit the proof of Theorem 5.2 here.

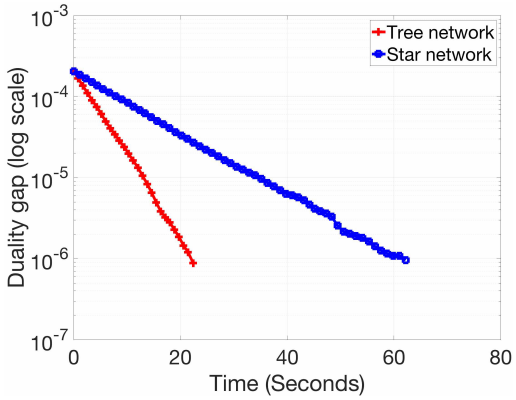

Fig. 3. Duality gap at center node as the operation time of algorithms goes. The distributed dual coordinate ascent in a tree network (red) and a star network (blue), i.e., CoCoA, are considered when the communication delay, t_{delay}, exists between the center node and its direct child nodes. $t_{delay} = 10^5 \times t_{lp}$, where t_{lp} represents the computational time for one local iteration at a worker, and the average $t_{lp} \approx 10^{-5}$. The number of local iteration, i.e., H, is set to 100.

6. NUMERICAL EXPERIMENTS

We demonstrate the convergence of the generalized distributed dual coordinate ascent in a tree network model. Since the authors in [5, 6] compared the distributed dual coordinate ascent in a star network, so-called CoCoA, with other methods including mini-batch SDCA [14], local SGD and mini-batch-SGD [15], we only compare our generalized distributed dual coordinate ascent in a tree network with the CoCoA [5]. Additionally, since we are interested in the distributed dual coordinate ascent considering different network topologies, we do not consider the CoCoA+ [6], which is the updated version of the CoCoA [5].

In the numerical experiment, we assume that lots of communication delays exist between the center node and local workers for the CoCoA [5]. For the generalized distributed dual coordinate ascent in a tree network, the same communication delays exist between the center node and the sub-center nodes (assuming that communication delays between sub-center nodes and local workers are negligible). We test our algorithm for the ridge regression problem with the wine quality dataset [16]. We consider a tree network model having four local workers, two sub-center nodes (each having two local workers), and one center node. The simulated star network has four local workers and one center node. In both cases, we evenly split the data to four local workers. Fig. 3 shows the duality gap at the center node as the operation time of the algorithms goes. We consider the case when the communication delay is 10^5 times larger than the local computational time for one local iteration at a local worker. As demonstrated in Fig. 3, the operation time of the distributed dual coordinate ascent can be further reduced by sharing local results via sub-center nodes when communication delays between the center node and its direct child nodes are large.

7. REFERENCES

[1] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[2] J. Chen, R. Monga, Bengio S., and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proceedings of the International Conference on Learning Representations Workshop Track*, 2016.

[3] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2011, pp. 69–77.

[4] T. Yang, "Trading computation for communication: Distributed stochastic dual coordinate ascent," in *Advances in Neural Information Processing Systems*, 2013, pp. 629–637.

[5] M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Advances in Neural Information Processing Systems*, 2014, pp. 3068–3076.

[6] C. Ma, V. Smith, M. Jaggi, M. I. Jordan, P. Richtárik, and M. Takáč, "Adding vs. averaging in distributed primal-dual optimization," in *Proceedings of the International Conference on Machine Learning*, 2015.

[7] S.-Y. Zhao and W.-J. Li, "Fast asynchronous parallel stochastic gradient descent: A lock-free approach with convergence guarantee," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[8] R. Zhang, S. Zheng, and J. T. Kwok, "Fast distributed asynchronous SGD with variance reduction," *CoRR, abs/1508.01633*, 2015.

[9] Z. Huo and H. Huang, "Distributed asynchronous dual free stochastic dual coordinate ascent," in *Proceedings of the IEEE International Conference on Data Mining*, 2018.

[10] C.-J. Hsieh, H.-F. Yu, and I. S. Dhillon, "PASSCoDe: Parallel asynchronous stochastic dual co-ordinate descent," in *Proceedings of the International Conference on Machine Learning*, 2015, vol. 15.

[11] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proceedings of the International Conference on Machine Learning*. ACM, 2008, pp. 408–415.

[12] K. Tsianos, S. Lawlor, and M. G. Rabbat, "Communication/computation tradeoffs in consensus-based distributed optimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 1943–1951.

[13] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *Journal of Machine Learning Research*, vol. 14, pp. 567–599, 2013.

[14] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro, "Mini-batch primal and dual methods for SVMs," in *Proceedings of the International Conference on Machine Learning*, 2013, pp. III–1022–III–1030.

[15] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: primal estimated sub-gradient solver for SVM," *Mathematical Programming*, vol. 127, no. 1, pp. 3–30, 2011.

[16] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems, Elsevier*, vol. 47, no. 4, pp. 547–553, 2009.