Convergence Rate of Distributed ADMM over Networks

Ali Makhdoumi and Asuman Ozdaglar MIT, Cambridge, MA 02139
Emails: makhdoum@mit.edu, asuman@mit.edu

***Abstract*—We propose a distributed algorithm based on Alternating Direction Method of Multipliers (ADMM) to minimize the sum of locally known convex functions using communication over a network. This optimization problem emerges in many applications in distributed machine learning and statistical estimation. We show that when functions are convex, both the objective function values and the feasibility violation converge with rate $O(\frac{1}{T})$, where T is the number of iterations. We then show that if the functions are strongly convex and have Lipschitz continuous gradients, the sequence generated by our algorithm converges linearly to the optimal solution. In particular, an ϵ-optimal solution can be computed with $O(\sqrt{\kappa_f}\log(1/\epsilon))$ iterations, where κ_f is the condition number of the problem. Our analysis also highlights the effect of network structure on the convergence rate through maximum and minimum degree of nodes as well as the algebraic connectivity of the network.**

I. Introduction

A. Motivation

Many of today's optimization problems in data science (including statistics, machine learning, and data mining) include an abundance of data, which cannot be handled by a single processor alone. This necessitates distributing data among multiple processors and processing it in a decentralized manner based on the available local information. The applications in machine learning [1], [2], [3], [4], [5] along with other applications in distributed data processing where information is inherently distributed among many processors (see e.g. distributed sensor networks [6], [7], coordination and flow control problems [8], [9]) have spearheaded a large literature on distributed multiagent optimization.

In this paper, we focus on the following optimization problem:

$$\min_{x\in\mathbb{R}^d} \sum_{i=1}^{n} f_i(x), \tag{1}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ is a convex function. We assume f_i is known only to agent i and refer to it as a local objective function.[1] Agents can communicate over a given network and their goal is to collectively solve this optimization. A prominent example where this general formulation emerges is *Empirical Risk Minimization* (EMR). Suppose that we have M data points $\{(x_i, y_i)\}_{i=1}^{M}$, where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \mathbb{R}$ is a target output. The empirical risk minimization is then given by

$$\min_{\theta\in\mathbb{R}^d} \frac{1}{M}\sum_{i=1}^{M} L(y_i, x_i, \theta) + p(\theta), \tag{2}$$

for some convex loss function $L : \mathbb{R}\times\mathbb{R}^d\times\mathbb{R}^d \to \mathbb{R}$ and some convex penalty function $p : \mathbb{R}^d \to \mathbb{R}$. This general formulation captures many statistical scenarios including:

- Least-Absolute Shrinkage and Selection Operator (LASSO):

$$\min_{\theta\in\mathbb{R}^d} \frac{1}{M}\sum_{i=1}^{M}(y_i - \theta' x_i)^2 + \tau||\theta||_1.$$

- Support Vector Machine (SVM) ([10]):

$$\min_{\theta\in\mathbb{R}^d} \frac{1}{M}\sum_{i=1}^{M}\max\{0, 1 - y_i(\theta' x_i)\} + \tau||\theta||_2^2.$$

Suppose our distributed computing system consists of n machines each with $k = M/n$ data points (without loss of generality suppose M is divisible by n, otherwise one of the machines have the remainder of data points). For all $i = 1, \ldots, n$, we define a function based on the available data to machine i as

$$f_i(\theta) = \frac{1}{k}\sum_{1+(i-1)k}^{ik} L(y_i, x_i, \theta) + p(\theta).$$

Therefore, the empirical risk minimization (2) can be written as $\min_{\theta\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$, where function $f_i(\theta)$ is only available to machine i, which is an instance of the formulation (1). Data is distributed across different machines either because it is collected by decentralized agents [11], [12], [13] or because memory constraints prevent it from being stored in a single machine[5], [14], [15], [16]. The decentralized nature of data together with communication constraints necessitate distributed processing which has motivated a large literature in optimization and statistical learning on distributed algorithms (see e.g. [17], [18], [19], [20], [21]).

B. Related Works and Contributions

Much of this literature builds on the seminal works [22], [23], which proposed gradient methods that can parallelize computations across multiple processors. A number of recent papers proposed subgradient type methods [24], [25], [26], [27], [28], [25], [29] or a dual averaging method [30] to design distributed optimization algorithms.

An alternative approach is to use Alternating Direction Method of Multipliers (ADMM) type methods which for separable problems leads to decoupled computations (see e.g. [31] and [32] for comprehensive tutorials on ADMM). ADMM has been studied extensively in the 80's [33], [34], [35]. More recently, it has found applications in a variety of distributed settings in machine learning such as model fitting, resource

[1] We use the terms machine, agent, and node interchangeably.

allocation, and classification (see e.g. [36], [37], [38], [39], [40], [41], [42], [43], [44]).

In this paper we present a new distributed ADMM algorithm for solving problem (1) over a network. Our algorithm relies on a novel node-based reformulation of (1) and leads to an ADMM algorithm that uses dual variables with dimension given by the number of nodes in the network. This results in a significant reduction in the number of variables stored and communicated with respect to edge-based ADMM algorithm presented in the literature (see [45], [46]). Our main contribution is a unified convergence rate analysis for this algorithm that applies to both the case when the local objective functions are convex and also the case when the local objective functions are strongly convex with Lipschitz continuous gradients. In particular, our analysis shows that when the local objective functions are convex (with no further assumptions), then the objective function at the ergodic average of the estimates generated by our algorithm converges with rate $O(\frac{1}{T})$. Moreover, when the local objective functions are strongly convex with Lipschitz continuous gradients we show that the iterates converges linearly, i.e., the iterates converge to an ϵ-neighborhood of the optimal solution after $O(\sqrt{\kappa_f}\log(\frac{1}{\epsilon}))$ steps, where κ_f is the condition number defined as L/ν, where L is the maximum Lipschitz gradient parameter and ν is the minimum strong convexity constant of the local objective functions. This matches the best known iteration complexity and condition number dependence for the centralized ADMM (see e.g. [47]). Our convergence rate estimates also highlight a novel dependence on the network structure as well as the communication weights. In particular, for communication weights that are governed by the Laplacian of the graph, we establish a novel iteration complexity $O\left(\sqrt{\kappa_f}\sqrt{\frac{d_{\text{max}}^4}{d_{\text{min}}a^2(G)}}\log\left(\frac{1}{\epsilon}\right)\right)$, where d_{min} is the minimum degree, d_{max} is the maximum degree, and $a(G)$ is the algebraic connectivity of the network. Finally, we illustrate the performance of our algorithm with numerical examples.

Our paper is most closely related to [45], [46], which studied edge-based ADMM algorithms for solving (1). In [46], the authors consider convex local objective functions and provide an $O(\frac{1}{T})$ convergence rate. The more recent paper [45] assumes strongly convex local objective functions with Lipschitz gradients and show a linear convergence rate through a completely different analysis. This analysis does not extend to the node-based ADMM algorithm under these assumptions. In contrast, our paper provides a unified convergence rate analysis for both cases for the node-based distributed ADMM algorithm. Our paper is also related to [48], [47] that study the basic centralized ADMM where the goal is to miminize sum of two functions with a linearly coupled constraint. Our work is also related to the literature on the converge of operator splitting schemes, such as Douglas-Rachford splitting and relaxed Peaceman-Rachford [49], [50], [51], [52], [53], [54], [55], [56], [57].

C. Outline

The organization of paper is as follows. In Section II we give the problem formulation and propose a novel distributed ADMM algorithm. In Section III, we show some preliminary results that helps us to show the main results. In Section IV we show the sub-linear convergence rate. In Section V we show the linear convergence rate of our algorithm. Finally, in Section VI we show the effect of network on the convergence rate and provide numerical results that illustrate the performance of our algorithm, which leads to concluding remarks in Section VII. All the omitted proofs are presented in the appendix.

II. Framework

A. Problem Formulation

Consider a network represented by a connected graph $G = (V, E)$ where $V = \{1, \dots, n\}$ is the set of agents and E is the set of edges. For any i, let $N(i)$ denote its set of neighbors including agent i itself, i.e., $N(i) = \{j \mid (i,j) \in E\} \cup \{i\}$, and let d_i denote the degree of agent i, i.e., $|N(i)| = d_i + 1$. We let $d_{\text{max}} = \max_{i\in V} d_i$ and $d_{\text{min}} = \min_{i\in V} d_i$.

The goal of the agents is to collectively solve optimization problem (1), where f_i is a function known only to agent i. In order to solve optimization problem (1), we introduce a variable x_i for each i and write the objective function of problem (1) as $\sum_{i=1}^n f_i(x_i)$, so that the objective function is decoupled across the agents. The constraint that all the x_i's are equal can be imposed using the following matrix.

Definition 1 (**Communication Matrix**). Let P be a $n \times n$ matrix whose entries satisfy the following property:
For any $i = 1, \dots, n$, $P_{ij} = 0$ for $j \notin N(i)$. We refer to P as the *communication matrix*.

Assumption 1. The communication matrix P satisfies $\text{null}(P) = \text{span}\{\mathbf{1}\}$, where $\mathbf{1}$ is a $n \times 1$ vector with all entries equal to one and $\text{null}(P)$ denotes the null-space of the matrix P.

Example 1. If $P_{ij} < 0$ for all $j \in N(i) \setminus \{i\}$, summation of each row of P is zero, and the graph is connected, then Assumption 1 holds. As a particular case, the Laplacian matrix of the graph given by $P_{ij} = -1$ when $j \in N(i) \setminus \{i\}$ and zero otherwise, and $P_{ii} = d_i$ is a communication matrix that satisfies Assumption 1.

We next show that the constraint that all x_i's are equal can be enforced by the linear constraint $A\mathbf{x} = 0$, where $\mathbf{x} = (x_1, \dots, x_n)$ where each x_i is a sub-vector of dimension d and A is a $dn \times dn$ matrix defined as the Kronecker product between communication matrix P and I_d, i.e., $A = P \otimes I_d$.

Lemma 1. *Under Assumption 1, the constraint* $A\mathbf{x} = 0$ *guarantees that* $x_i = x_j$ *for all* $i, j \in V$.

Using Lemma 1, under Assumption 1, we can reformulate optimization problem (1) as

$$\min_{\mathbf{x}\in\mathbb{R}^{nd}} F(\mathbf{x}) \quad \text{s.t. } A\mathbf{x} = 0, \tag{3}$$

where $F(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$.

Assumption 2. The optimal solution set of problem (3) is non-empty. We let $\mathbf{x}^*$ denote an optimal solution of the problem (3).

B. Multiagent Distributed ADMM

In this section, we propose a distributed ADMM algorithm to solve problem (3). We first use a reformulation technique (this technique was introduced in [58] to separate optimization variables in a constraint, allowing them to be updated simultaneously in an ADMM iteration), which allows us to separate each constraint associated with a node into multiple constraints that involve only the variable corresponding to one of the neighboring nodes. We expand the constraint $A\mathbf{x} = 0$ so that for each node i, we have $\sum_{j\in N(i)} A_{ij}x_j = 0$, where $A_{ij} = P_{ij} \otimes I_d$ is a $d \times d$ matrix. We let $A_{ij}x_j = z_{ij} \in \mathbb{R}^d$ to obtain the following reformulation:

$$\begin{aligned} &\min_{\mathbf{x},\ \mathbf{z}} F(\mathbf{x}) \\ &\text{s.t. } A_{ij}x_j = z_{ij}, \quad \text{for } i = 1, \dots, n,\ j \in N(i), \\ &\qquad \sum_{j\in N(i)} z_{ij} = 0, \quad \text{for } i = 1, \dots, n. \end{aligned} \tag{4}$$

For each equality constraint in (4), we let $\lambda_{ij} \in \mathbb{R}^d$ be the corresponding Lagrange multiplier and form the augmented Lagrangian function by adding a quadratic penalty with penalty parameter $c > 0$ for feasibility violation to the Lagrangian function as

$$\begin{aligned} L_c(\mathbf{x}, \mathbf{z}, \lambda) = F(\mathbf{x}) &+ \sum_{i=1}^{n} \sum_{j\in N(i)} \lambda'_{ij}(A_{ij}x_j - z_{ij}) \\ &+ \frac{c}{2} \sum_{i=1}^{n} \sum_{j\in N(i)} ||A_{ij}x_j - z_{ij}||_2^2. \end{aligned}$$

We now use ADMM algorithm (see e.g. [59]). ADMM algorithm generates primal-dual sequences $\{x_j(t)\}, \{z_{ij}(t)\}$, and $\{\lambda_{ij}(t)\}$ which at iteration $t+1$ are updated as follows:

1) For any $j = 1, \dots, n$, we update x_j as
$$x_j(t+1) \in \text{argmin}_{x_j\in\mathbb{R}^d} L_c(\mathbf{x}, \mathbf{z}(t), \lambda(t)). \tag{5}$$
2) For any $i = 1, \dots, n$, we update the vector $\mathbf{z}_i = [z_{ij}]_{j\in N(i)}$ as
$$\mathbf{z}_i(t+1) \in \text{argmin}_{\mathbf{z}_i\in Z_i} L_c(\mathbf{x}(t+1), \mathbf{z}, \lambda(t))., \tag{6}$$
where $Z_i = \{\mathbf{z}_i \mid \sum_{j\in N(i)} z_{ij} = 0\}$.
3) For $i = 1, \dots, n$ and $j \in N(i)$ we update λ_{ij} as
$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + c(A_{ij}x_j(t+1) - z_{ij}(t+1)). \tag{7}$$

One can implement this algorithm in a distributed manner, where node i maintains variables $\lambda_{ij}(t)$ and $z_{ij}(t)$ for all $j \in N(i)$ ([46]). However, using the inherent symmetries in the problem, we can significantly reduce the number of variables that each node requires to maintain from $O(|E|)$ to $O(|V|)$.

We first show that for all t, i, and $j \in N(i)$, we have $\lambda_{ij}(t) = p_i(t)$. This reduction shows that the algorithm need not maintain dual variables $\lambda_{ij}(t)$ for each i and its neighbors j, but instead can operate with the lower dimensional node-based dual variable $p_i(t)$. The dual variable $p_i(t)$ can be updated using primal variables $x_j(t)$ for all $j \in N(i)$. The second observation is that $z_{ij}(t) = A_{ij}x_j(t) - y_i(t)$, where $y_i(t) = \frac{1}{d_i+1}\left([A]^i\right)' \mathbf{x}(t)$ and $\left([A]^i\right)' = (P_{i1}, \dots, P_{in}) \otimes I_d$. This reduction shows that the algorithm need not maintain primal variables $z_{ij}(t)$ for each i and its neighbors j, but instead can operate with the lower dimensional node-based primal variables $y_i(t)$, where $y_i(t)$ is node i's estimate of the primal variable (obtained as the average of primal variables of his own neighbors). The aforementioned reductions are shown in the following proposition.

Algorithm 1 Multiagent Distributed ADMM

```
• Initialization: $x_i(0)$, $y_i(0)$, and $p_i(0)$ all in $\mathbb{R}^d$, for any $i \in V$ and matrix $A \in \mathbb{R}^{nd\times nd}$.
• Algorithm:
    1) for $i = 1, \dots, n$, let
        $x_i(t+1) \in \text{argmin}_{x_i\in\mathbb{R}^d} f_i(x_i) + \sum_{j\in N(i)} (p'_j(t)A_{ji}x_i + \frac{c}{2}||y_j(t) + A_{ji}(x_i - x_i(t)||_2^2).$
    2) for $i = 1, \dots, n$, let
        $y_i(t+1) = \frac{1}{d_i+1} \sum_{j\in N(i)} A_{ij}x_j(t+1).$
    3) for $i = 1, \dots, n$, let
        $p_i(t+1) = p_i(t) + cy_i(t+1)$
• Output: $\{x_i(t)\}_{t=0}^{\infty}$ for any $i \in V$.
```

Proposition 1. *The sequence $\{x_i(t)\}_{t=0}^{\infty}$ for $i = 1, \dots, n$ generated by implementing the steps presented in Algorithm* (1) *is the same as the sequence generated by the ADMM algorithm.*

The steps of the algorithm can be implemented in a distributed way, meaning that each node first updates her estimates based on the information received from her neighboring nodes and then broadcasts her updated estimates to her neighboring nodes. Each node i maintains local variables $x_i(t)$, $y_i(t)$, and $p_i(t)$ and updates these variables using communication with its neighbors as follows:

- At the end of iteration t, each node i sends out $p_i(t)$ and $y_i(t)$ to all of its neighbors and then each node such as j uses $y_i(t)$ and $p_i(t)$ of all $i \in N(j)$ to update $x_j(t+1)$ as in step 1.
- Each node j sends out $x_j(t+1)$ to all of its neighbors and then each node such as i computes $y_i(t+1)$ as in step 2.
- Each node i updates $p_i(t+1)$ as in step 3.

Using this algorithm agent i need to store only three variables, $x_i(t)$, $y_i(t)$, and $p_i(t)$ and update them at each iteration. Also, each agent need to communicate only with (broadcast her estimates to) its neighbors. Therefore, the overall storage requirement is $3|V|$ and the overall communication requirement at each iteration is $|E|$.

III. Preliminary Results

In this section, we present the preliminary results that we will use to establish our convergence rate. we define

$$\partial F(\mathbf{x}) = \{h \in \mathbb{R}^{nd} \; : \; h = (h_1(x_1)', \ldots, \nabla h_n(x_n)')' \\ , \; h_i(x_i) \in \partial f_i(x_i)\},$$

where for each i, $\partial f_i(x_i)$ denotes subdifferential of f_i at x_i, i.e., the set of all subgradients of f_i at x_i. In what follows, for notational simplicity we assume $d = 1$, i.e., in (3) $x \in \mathbb{R}$. All the analysis generalizes to the case with $x \in \mathbb{R}^d$. We first provide a compact representation of the evolution of primal vector $\mathbf{x}(t)$ that will be used in the convergence proof. This is a core step in proving the convergence rate as it eliminates the dependence on the other variables $y_i(t)$ and $p_i(t)$. Let M be a $n \times n$ diagonal matrix with $M_{ii} = \sum_{j \in N(i)} A_{ji}^2$ and D be a $n \times n$ diagonal matrix with $D_{ii} = d_i + 1$.

Lemma 2 (**Perturbed Linear Update**). *The update of Algorithm 1 can be written as*

$$\mathbf{x}(t+1) = -\frac{1}{c} M^{-1} h(\mathbf{x}(t+1)) + \left(I - M^{-1} A' D^{-1} A\right) \mathbf{x}(t) \\ - M^{-1}(A' D^{-1} A) \sum_{s=0}^{t} \mathbf{x}(s),$$

for some $h(\mathbf{x}(t+1)) \in \partial F(\mathbf{x}(t+1))$.

Lemma 2 shows $\mathbf{x}(t+1)$ can be written as a perturbed linear combination of $\{\mathbf{x}(s)\}_{s=0}^{t}$ with the perturbation being the term $-\frac{1}{c} M^{-1} h(\mathbf{x}(t+1))$. The intuition behind the convergence rate analysis is that the linear term that relates $\mathbf{x}(t+1)$ to $\mathbf{x}(0), \ldots, \mathbf{x}(t)$ guarantees that the sequence $\mathbf{x}(t)$ converges to a consensus point where $x_i(t) = x_j(t)$ for all $i, j \in V$; and the perturbation term $-\frac{1}{c} M^{-1} h(\mathbf{x}(t+1))$ guarantees that the converging point minimizes the objective function $F(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i)$.

IV. Sub-linear Rate of Convergence

In this section, we show the sublinear rate of convergence. We define two auxiliary sequences that we will use in proving the convergence rates. Since $A' D^{-1} A$ is positive semidefinite (see Lemma 6 in the appendix), we can define $Q = (A' D^{-1} A)^{1/2}$. In other words, we let $Q = V \Sigma^{1/2} V'$, where $A' D^{-1} A = V \Sigma V'$ is the singular value decomposition of the symmetric matrix $A' D^{-1} A$. We define the auxiliary sequences

$$\mathbf{r}(t) = \sum_{s=0}^{t} Q \mathbf{x}(s),$$

and

$$\mathbf{q}(t) = \begin{pmatrix} \mathbf{r}(t) \\ \mathbf{x}(t) \end{pmatrix}.$$

We also let

$$G = \begin{pmatrix} I & 0 \\ 0 & M - A' D^{-1} A \end{pmatrix}.$$

Next, we show a proposition that bounds the function value at each iteration.

Proposition 2. *For any $\mathbf{r} \in \mathbb{R}^d$ and t, the sequence generated by Algorithm 1 satisfies:*

$$\frac{2}{c}\left(F(\mathbf{x}(t+1)) - F(\mathbf{x}^*)\right) + 2\mathbf{r}' Q \mathbf{x}(t+1) \\ \leq ||\mathbf{q}(t) - \mathbf{q}^*||_G^2 - ||\mathbf{q}(t+1) - \mathbf{q}^*||_G^2 - ||\mathbf{q}(t) - \mathbf{q}(t+1)||_G^2,$$

where $\mathbf{q}^* = \begin{pmatrix} \mathbf{r}^* \\ \mathbf{x}^* \end{pmatrix}$.

In order to obtain $O(1/T)$ convergence rate, we consider the performance of the algorithm at the ergodic vector defined as $\hat{\mathbf{x}}(T) = (\hat{x}_1(T), \ldots, \hat{x}_n(T))$, where

$$\hat{x}_i(T) = \frac{1}{T} \sum_{t=1}^{T} x_i(t),$$

for any $i = 1, \ldots, n$. Note that each agent i can construct this vector by simple recursive time-averaging of its estimate $x_i(t)$. Let $(\mathbf{x}^*, \hat{\mathbf{r}})$ be a primal-dual optimal solution of

$$\min_{Q\mathbf{x}=0} F(\mathbf{x}).$$

Since null(Q) = null(P), under Assumption (1), the optimal primal solution of this problem is the same as of the original problem (3) Next, we show both objective function and feasibility violation converges with rate $O(\frac{1}{T})$ to the optimal value.

Theorem 1. *For any T, we have*

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||_{M - A'D^{-1}A}^2\right) \\ + \frac{c}{2T}\left(\max\{||\mathbf{r}(0) - 2\hat{\mathbf{r}}||_2^2, ||\mathbf{r}(0)||_2^2\}\right).$$

We also have

$$||Q\hat{\mathbf{x}}(T)||_2 \leq \frac{1}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||_{M - A'D^{-1}A}^2\right) \\ + \frac{1}{2T}\left(2||\mathbf{r}(0) - \hat{\mathbf{r}}||_2^2 + 2\right).$$

This theorem shows that the objective function at the ergodic average of the sequence of estimates generated by Algorithm 1 converges with rate $O(\frac{1}{T})$ to the optimal solution. We next characterize the network effect on the performance guarantee.

Theorem 2. *For any T, starting form $\mathbf{x}(0) = 0$, we have*

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}||\mathbf{x}^*||_2^2 \lambda_M + \frac{2}{cT}\frac{U^2}{\tilde{\lambda}_m},$$

and

$$||Q\hat{\mathbf{x}}(T)|| \leq \frac{1}{2T}||\mathbf{x}^*||_2^2 \lambda_M + \frac{1}{2T}\left(2 + 2\frac{U^2}{c^2 \tilde{\lambda}_m}\right),$$

where U is a bound on the subgradients of the function F at $\mathbf{x}^*$, i.e., $\|\mathbf{v}\| \leq U$ for all $\mathbf{v} \in \partial F(\mathbf{x}^*)$, $\tilde{\lambda}_m$ is the smallest non-zero eigen value of $A'D^{-1}A$, and λ_M is the largest eigen value of $M - A'D^{-1}A$.

Remark 1. Both the optimality of the objective function value at the ergodic average and the feasibility violation converge with rate $O(\frac{1}{T})$. Our guaranteed rates show a novel dependency on the network structure and communication matrix through $\tilde{\lambda}_m$ and λ_M. Therefore, for a given function, in order to obtain a better performance guarantee we need to maximize $\tilde{\lambda}_m$ and minimize λ_M. In Section (VI) we show that these terms depend on the algebraic connectivity of the network and provide explicit dependencies solely on the network structure when the communication matrix is the Laplacian of the graph.

V. Linear Rate of Convergence

In order to show the linear rate of convergence, we adopt the following standard assumptions.

Assumption 3 (Strongly convex and Lipschitz Gradient). For any $i = 1, \dots, n$, the function f_i is differentiable and has Lipschitz continuous gradient, i.e.,

$$|\nabla f_i(x) - \nabla f_i(y)| \leq L_{f_i} ||x - y||_2, \text{ for any } x, y \in \mathbb{R}^d,$$

for some $L_{f_i} \geq 0$. The function f_i is also strongly convex with parameter $\nu_{f_i} > 0$, i.e., $f_i(x) - \frac{\nu_{f_i}}{2}||x||_2^2$ is convex.

We let $\nu = \min_{1 \leq i \leq n} \nu_{f_i}$ and $L = \max_{1 \leq i \leq n} L_{f_i}$, and define the condition number of $F(\mathbf{x})$ (or the condition number of problem (3)) as $\kappa_f = \frac{L}{\nu}$. Note that when the functions are differentiable, we have

$$\nabla F(\mathbf{x}) = (\nabla f_1(x_1)', \dots, \nabla f_n(x_n)')' \in \mathbb{R}^{nd}.$$

Assumption 3 results in the following standard inequalities for the aggregate function $F(\mathbf{x})$.

Lemma 3. (a) *Under Assumption 3, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{nd}$, we have*

$$(\nabla F(\mathbf{x}) - \nabla F(\mathbf{y}))'(\mathbf{x} - \mathbf{y}) \geq \nu ||\mathbf{x} - \mathbf{y}||_2^2.$$

(b) *Under Assumption 3, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{nd}$, we have*

$$(\nabla F(\mathbf{x}) - \nabla F(\mathbf{y}))'(\mathbf{x} - \mathbf{y}) \geq \frac{1}{L} ||\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})||_2^2.$$

(c) *Under convexity assumption, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{nd}$ and $h(\mathbf{x}) \in \partial F(\mathbf{x})$ we have*

$$(\mathbf{x} - \mathbf{y})' h(\mathbf{x}) \geq F(\mathbf{x}) - F(\mathbf{y}).$$

Under Assumption (3) we show that the sequence generated by Algorithm (1) converges linearly to the optimal solution (which is unique under these assumptions). The idea is to use strong convexity and Lipschitz gradient property of $F(\mathbf{x})$ in order to show that the G-norm of sequence $\mathbf{q}(t) - \mathbf{q}^*$ contracts at each iteration, providing a linear rate.

Theorem 3. *Suppose Assumptions 1, 2, and 3 hold. For any value of the penalty parameter $c > 0$ and $\beta \in (0, 1)$, the sequence generated by Algorithm 1 $\{\mathbf{x}(t)\}_{t=1}^{\infty}$ satisfies*

$$||\mathbf{x}(t) - \mathbf{x}^*||_2^2 \leq \left(\frac{1}{1 + \delta}\right)^t ||\mathbf{q}(0) - \mathbf{q}^*||_2^2,$$

where

$$\delta \leq \min\left\{\frac{2\beta\nu}{c\lambda_M(1 + \frac{2}{\tilde{\lambda}_m})}, \frac{(1 - \beta)c\tilde{\lambda}_m}{L}\right\},$$

and $\tilde{\lambda}_m$ is the smallest non-zero eigen value of $A'D^{-1}A$, λ_M is the largest eigen value of $M - A'D^{-1}A$.

The rate of convergence in Theorem 3 holds for any choice of penalty parameter $c > 0$. In other words, for any choice of $c > 0$, the convergence rate is linear. We now optimize the rate of convergence over all choices of c and provide an explicit convergence rate estimate that highlights dependence on the condition number of the problem.

Theorem 4. *Suppose Assumptions 1, 2, and 3 hold. Let $\{\mathbf{x}(t)\}_{t=1}^{\infty}$ be the sequence generated by Algorithm 1. There exist $c > 0$ for which we have*

$$||\mathbf{x}(t) - \mathbf{x}^*||_2^2 \leq \rho^t ||\mathbf{q}(0) - \mathbf{q}^*||_2^2,$$

where the rate $\rho < 1$ is given by

$$\rho = \left(1 + \frac{1}{2}\sqrt{\frac{2\tilde{\lambda}_m^2}{\lambda_M(2 + \tilde{\lambda}_m)}\frac{1}{\kappa_f}}\right)^{-1}.$$

Remark 2. This result shows that within $O\left(\sqrt{\kappa_f}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations, the estimates $\{\mathbf{x}(t)\}$ reach an ϵ- neighborhood of the optimal solution. Our rate estimate has a $\sqrt{\kappa_f}$ dependence which improves on the linear condition number dependence provided in the convergence analysis of edge-based ADMM in [47]. The network dependence in our rate estimates is captured through $\tilde{\lambda}_m$ and λ_M. In particular, the larger $\tilde{\lambda}_m$ and the smaller λ_M results in a faster rate of convergence. In Section (VI) we will explicitly show the network effect in the convergence rate and provide numerical results that illustrate the performance for networks with different connectivity properties.

VI. Network Effects

We can choose communication matrix P (and the corresponding matrix A) in the Algorithm 1 to be any matrix that satisfies Assumption (1). One natural choice for the matrix P is the Laplacian of the graph which leads to having $A_{ij} = A_{ji} = -1$ for all $j \in N(i) \setminus \{i\}$ and $A_{ii} = d_i$. Using Laplacian as the communication matrix we can now capture the effect of network structure in the convergence rate.

A. Network Effect in Sub-linear Rate

The following proposition explicitly show the networks dependence in the bounds provided in Theorem 2.

Proposition 3. *For any T, starting form $\mathbf{x}(0) = 0$ and using standard Laplacian as the communication matrix, we have*

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}||\mathbf{x}^*||_2^2\left(4d_{max}^2\right) + \frac{2}{cT}U^2\frac{2d_{max}}{a(G)^2},$$

and

$$||Q\hat{\mathbf{x}}(T)|| \leq \frac{1}{2T}||\mathbf{x}^*||_2^2\left(4d_{max}^2\right) + \frac{1}{2T}\left(2 + \frac{2U^2}{c^2}\frac{2d_{max}}{a(G)^2}\right),$$

where U is a bound on the subgradients of the function F at $\mathbf{x}^*$, i.e., $\|\mathbf{v}\| \leq U$ for all $\mathbf{v} \in \partial F(\mathbf{x}^*)$ and $a(G)$ is the algebraic connectivity of the graph.

Therefore, highly connected graphs with larger algebraic connectivity has a faster convergence rate (see e.g. [60], [61] for an overview of the results on algebraic connectivity).

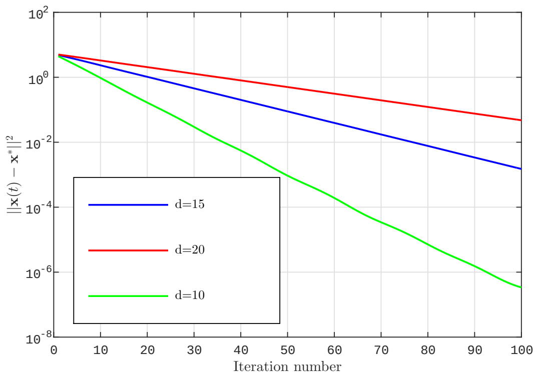

Fig. 1: Performance of Algorithm 1 for three d-regular graphs with $d = 10, 20, 30$. The y axis is logarithmic to show the linear convergence rate.

B. Network Effect in Linear Rate

The following proposition explicitly show the networks dependence in the bound provided in Theorem 4.

Proposition 4. *Suppose Assumptions 1, 2, and 3 hold. Using standard Laplacian as the communication matrix, in order to reach an* ϵ*-optimal solution* $O\left(\sqrt{\kappa_f}\sqrt{\frac{d_{max}^4}{d_{min}a^2(G)}}\log\left(\frac{1}{\epsilon}\right)\right)$ *iterations suffice.*

Both of our guaranteed rates for sub-linear and linear rates depends on three parameters $d_{\max}$, $d_{\min}$ and $a(G)$. The convergence rate is faster for larger $d_{\min}$ and smaller $d_{\max}$. Finally, the convergence rate is faster for larger algebraic connectivity $a(G)$.

Example 2. To provide more intuition on the networks dependence, we focus on d-regular graphs with matrix P equal to Laplacian of the graph. In this setting, we have: $\tilde{\lambda}_m = \frac{a(G)^2}{d+1}$ and $\lambda_M = d(d+1)$, where $a(G)$ is the algebraic connectivity of the graph. Thus in this case the iteration complexity is $O\left(\sqrt{\kappa_f}\sqrt{\frac{d^3}{a^2(G)}}\log\left(\frac{1}{\epsilon}\right)\right)$ (note that this bound matches the one provided in Proposition 4). For d-regular graphs there exist good expanders such as Ramanujan graphs for which $a(G) = O(d)$ (see e.g. [62]). In Figure 1, we compare the performance of our algorithm for several regular graphs. The choice of function is $F(x) = \frac{1}{2}\sum_{i=1}^n (x - a_i)^2$ where a_i is a scalar that is known only to machine i (where $a_i = i$ for $i = 1, \ldots, n$). The communication matrix used in these experiments is the Laplacian of the graph. This problem appears in distributed estimation where the goal is to estimate the parameter x^*, using local measurements $a_i = x^* + N_i$ at each machine $i = 1, \ldots, n$. Here N_i represents measurements noise, which we assume to be jointly Gaussian with mean zero and variance one. The maximum likelihood estimate is the minimizer x^* of $F(x)$.

VII. CONCLUSION

We proposed a novel distributed algorithm based on Alternating Direction Method of Multipliers (ADMM) to minimize the sum of locally known convex functions. We first showed that ADMM can be implemented by only keeping track of some node-based variables. We then showed that our algorithm reaches ϵ-optimal solution in $O\left(\frac{1}{\epsilon}\right)$ number of iterations for convex functions and in $\left(\sqrt{\kappa_f}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations for strongly convex and Lipschitz functions. Our analysis shows that the performance of our algorithm depends on the algebraic connectivity of the graph, the minimum degree of the nodes, and the maximum degree of the nodes. Finally, we illustrated the performance of our algorithm with numerical examples.

APPENDIX

A. **Proof of Lemma 1**

Consider the k-th coordinate of all x_i's and form a $n \times 1$ vector $\mathbf{x}^k$. From $A\mathbf{x} = 0$ and the fact that $A = P \otimes I$, we obtain that $P\mathbf{x}^k = 0$. This shows that $\mathbf{x}^k \in \text{null}(P)$. Using Assumption 1, we have $\mathbf{x}^k \in \text{span}(\{\mathbf{1}\})$, which guarantees that all entries of $\mathbf{x}^k$ are equal. Similarly, for any $k = 1, \ldots, d$, the k-th entries of all x_i's are equal. This leads to $x_i = x_j$ for all $i, j \in V$.

B. **Proof of Lemma 2**

Using the first step of Algorithm (1) for i, we can write $x_i(t+1)$ as

$$h_i(\mathbf{x}(t+1)) + \sum_{j \in N(i)} A_{ji}p_j(t)$$
$$+ c\sum_{j \in N(i)} A_{ji}(y_j(t) + A_{ji}x_i(t+1) - A_{ji}x_i(t)) = 0, \quad (8)$$

where $h_i(\mathbf{x}(t+1)) = f_i'(x_i(t+1))$ for differentiable functions and $h_i(\mathbf{x}(t+1)) \in \partial f_i(x_i(t+1))$ in general. We next use second and third steps of Algorithm (1) to write $p_i(t)$ and $y_i(t)$ in terms of $(\mathbf{x}(0), \ldots, \mathbf{x}(t))$. Using the update for j, we have

$$\sum_{j \in N(i)} p_j(t)A_{ji} = \sum_{j \in N(i)} A_{ji}\sum_{s=0}^{t} \frac{c}{d_i + 1}\sum_{s=0}^{t}([A]^j)'\mathbf{x}(s)$$
$$= c\sum_{s=0}^{t}[A'D^{-1}Ax(s)]_i. \quad (9)$$

Moreover, we can write the term $\sum_{j \in N(i)} A_{ji}y_j(t)$ based on the sequence $(\mathbf{x}(0), \ldots, \mathbf{x}(t))$ as follows

$$\sum_{j \in N(i)} A_{ji}y_j(t) = \sum_{j \in N(i)} A_{ji}\frac{1}{d_j + 1}([A]^j)'\mathbf{x}(t)$$
$$= [A'D^{-1}A\mathbf{x}(t)]_i. \quad (10)$$

Substituting (9) and (10) in (8), we can write the update of $x_i(t+1)$ in terms of the sequence $(\mathbf{x}(0), \ldots, \mathbf{x}(t))$, which then can compactly be written as

$$cM\mathbf{x}(t+1) = -h(\mathbf{x}(t+1)) + c\left(M - A'D^{-1}A\right)\mathbf{x}(t)$$
$$- c(A'D^{-1}A)\sum_{s=0}^{t}\mathbf{x}(s),$$

where $h(\mathbf{x}(t+1)) = \nabla F(\mathbf{x}(t+1))$ if the functions are differentiable and $h(\mathbf{x}(t+1)) \in \partial F(\mathbf{x}(t+1))$ in general. Left multiplying by $\frac{1}{c}M^{-1}$, completes the proof.

C. Proof of Proposition 2

We first show a lemma that we will use in the proof of this proposition. The following lemma shows the relation between the auxiliary sequence $\mathbf{r}(t)$ and the primal sequence $\mathbf{x}(t)$.

Lemma 4. *Suppose Assumptions 1 and 2 hold. The sequence* $\{\mathbf{x}(t), \mathbf{r}(t)\}_{t=0}^{\infty}$ *satisfies*

$$\begin{aligned}&(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))\\ &= -Q\mathbf{r}(t+1) - \frac{1}{c}h(\mathbf{x}(t+1)),\end{aligned}$$

for some $h(\mathbf{x}(t+1)) \in \partial F(\mathbf{x}(t+1))$.

Proof: Using Lemma 2 we have

$$\begin{aligned}&M(\mathbf{x}(t+1) - \mathbf{x}(t)) = -\frac{1}{c}h(\mathbf{x}(t+1))\\ &- (A'D^{-1}A)\mathbf{x}(t) - (A'D^{-1}A)\sum_{s=0}^{t}\mathbf{x}(s).\end{aligned}$$

We subtract $(A'D^{-1}A)\mathbf{x}(t+1)$ from both sides and rearrange the terms to obtain

$$\begin{aligned}&(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))\\ &= -A'D^{-1}A\sum_{s=0}^{t+1}\mathbf{x}(s) - \frac{1}{c}h(\mathbf{x}(t+1)).\end{aligned}$$

Using $QQ = A'D^{-1}A$, yields

$$\begin{aligned}&(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))\\ &= -Q\mathbf{r}(t+1) - \frac{1}{c}h(\mathbf{x}(t+1)).\end{aligned}$$

■

Back to the proof of Proposition 2: Using Lemma 7 and Lemma 3 part (c), we have that

$$\begin{aligned}&\frac{2}{c}\left(F(\mathbf{x}(t+1)) - F(\mathbf{x}^*)\right) + 2\mathbf{r}'Q\mathbf{x}(t+1\\ &\leq \frac{2}{c}(\mathbf{x}(t+1) - \mathbf{x}^*)'h(\mathbf{x}(t+1)) + 2\mathbf{r}'Q\mathbf{x}(t+1)\\ &= 2(\mathbf{x}(t+1) - \mathbf{x}^*)'(-Q(\mathbf{r}(t+1) - \mathbf{r})\\ &- (M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t)))\\ &= 2(\mathbf{r}(t+1) - \mathbf{r}(t))'\left(-\mathbf{r}(t+1) + \mathbf{r}\right)\\ &- 2(\mathbf{x}(t+1) - \mathbf{x}^*)'(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))\\ &= \left(||\mathbf{r}(t) - \mathbf{r}^*||_2^2 - ||\mathbf{r}(t+1) - \mathbf{r}||_2^2 - ||\mathbf{r}(t) - \mathbf{r}(t+1)||_2^2\right)\\ &+ (||\mathbf{x}(t) - \mathbf{x}^*||^2_{(M-A'D^{-1}A)} - ||\mathbf{x}(t+1) - \mathbf{x}^*||^2_{(M-A'D^{-1}A)}\\ &- ||\mathbf{x}(t) - \mathbf{x}(t+1)||^2_{(M-A'D^{-1}A)})\\ &= ||\mathbf{q}(t) - \mathbf{q}^*||_G^2 - ||\mathbf{q}(t+1) - \mathbf{q}^*||_G^2 - ||\mathbf{q}(t) - \mathbf{q}(t+1)||_G^2.\end{aligned}$$

D. Proof of Theorem 1

Taking summation of the relation given in Proposition 2 from $t = 0$ up to $t = T$, we obtain that

$$\sum_{t=0}^{T} F(\mathbf{x}(t)) - F(\mathbf{x}^*) + c\mathbf{r}'Q\mathbf{x}(t) \leq \frac{c}{2}||\mathbf{q}(0) - \mathbf{q}||_G^2.$$

Using convexity of the functions and Jensen's inequality, we obtain

$$F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + c\mathbf{r}'Q\hat{\mathbf{x}}(T) \leq \frac{c}{2T}||\mathbf{q}(0) - \mathbf{q}||_G^2.$$

Letting $\mathbf{r} = 0$, yields

$$\begin{aligned}&F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)\\ &\leq \frac{c}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A} + ||\mathbf{r}(0)||_2^2\right).\end{aligned} \tag{11}$$

From saddle point inequality, we have

$$F(\mathbf{x}^*) \leq F(\hat{\mathbf{x}}(T)) + c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T), \tag{12}$$

which implies

$$F(\mathbf{x}^*) - F(\hat{\mathbf{x}}(T)) \leq c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T). \tag{13}$$

Next, we will bound the term $\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T)$. We add the term $c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T)$ to both sides of (12) to obtain

$$c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T) \leq F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + 2c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T). \tag{14}$$

Again, using Proposition 2 to bound the right-hand side of (14), we obtain

$$c\mathbf{r}^{*\prime}Q\hat{\mathbf{x}}(T) \leq \frac{c}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A} + ||\mathbf{r}(0) - 2\hat{\mathbf{r}}||_2^2\right). \tag{15}$$

Using (15) to bound the right-hand side of (13), and then combining the result with (12), we obtain

$$\begin{aligned}&|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A}\right)\\ &+ \frac{c}{2T}\left(\max\{||\mathbf{r}(0) - 2\hat{\mathbf{r}}||_2^2, ||\mathbf{r}(0)||_2^2\}\right).\end{aligned}$$

We next bound the feasibility violation. Using Proposition 2 with $\mathbf{r} = \hat{\mathbf{r}} + \frac{Q\hat{\mathbf{x}}(T)}{||Q\hat{\mathbf{x}}(T)||}$, we have

$$\begin{aligned}&F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + c\mathbf{r}^{*\prime}Q\hat{\mathbf{x}}(T) + c||Q\hat{\mathbf{x}}(T)||\\ &\leq \frac{c}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A}\right)\\ &+ \frac{c}{2T}\left(||\mathbf{r}(0) - \hat{\mathbf{r}} - \frac{Q\hat{\mathbf{x}}(T)}{||Q\hat{\mathbf{x}}(T)||}||_2^2\right).\end{aligned}$$

Since $(\mathbf{x}^*, \hat{\mathbf{r}})$ is a primal-dual optimal solution, using saddle point inequality, we have that

$$F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*) + c\hat{\mathbf{r}}'Q\hat{\mathbf{x}}(T) \geq 0.$$

Combining the two previous relations, we obtain

$$\begin{aligned}&||Q\hat{\mathbf{x}}(T)||_2 \leq \frac{1}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A}\right)\\ &+ \frac{1}{2T}\left(||\mathbf{r}(0) - \hat{\mathbf{r}} - \frac{Q\hat{\mathbf{x}}(T)}{||Q\hat{\mathbf{x}}(T)||}||_2^2\right).\end{aligned}$$

Since

$$||\mathbf{r}(0) - \hat{\mathbf{r}} - \frac{Q\hat{\mathbf{x}}(T)}{||Q\hat{\mathbf{x}}(T)||}||_2^2 \leq 2||\mathbf{r}(0) - \hat{\mathbf{r}}||_2^2 + 2,$$

we can further bound $||Q\hat{\mathbf{x}}(T)||$ as

$$\begin{aligned}&||Q\hat{\mathbf{x}}(T)||_2 \leq \frac{1}{2T}\left(||\mathbf{x}(0) - \mathbf{x}^*||^2_{M-A'D^{-1}A}\right)\\ &+ \frac{1}{2T}\left(2||\mathbf{r}(0) - \hat{\mathbf{r}}||_2^2 + 2\right).\end{aligned}$$

E. Proof of Theorem 2

We first show a lemma that bounds the norm of dual optimal solution of (16).

Lemma 5. *Let $\mathbf{x}^*$ be an optimal solution for problem* (3)*. There exists an optimal dual solution $\tilde{\mathbf{r}}$ for problem*

$$\min_{cQ\mathbf{x}=0} F(\mathbf{x}), \tag{16}$$

that satisfies

$$||\tilde{\mathbf{r}}||_2^2 \leq \frac{U^2}{c^2\tilde{\lambda}_m},$$

where U is a bound on the subgradients of the function F at $\mathbf{x}^*$, i.e., $\|\mathbf{v}\| \leq U$ for all $\mathbf{v} \in \partial F(\mathbf{x}^*)$, and $\tilde{\lambda}_m$ is the smallest non-zero eigen-value of $A'D^{-1}A$.

Proof: There exists an optimal primal-dual solution for problem (16) such that $(\mathbf{x}^*, \hat{\mathbf{r}})$ is a saddle point of the Lagrangian function, i.e., for any $\mathbf{x} \in \mathbb{R}^n$,

$$F(\mathbf{x}^*) - F(\mathbf{x}) \leq c\hat{\mathbf{r}}'Q(\mathbf{x} - \mathbf{x}^*). \tag{17}$$

Note that $(\mathbf{x}^*, \hat{\mathbf{r}})$ satisfies saddle point inequality if and only if it satisfies the inequality given in (17). Equation (17) shows that $c\hat{\mathbf{r}}'Q \in \partial F(\mathbf{x}^*)$. Let $c\hat{\mathbf{r}}'Q = \mathbf{v}' \in \partial F(\mathbf{x}^*)$. We will use this $\hat{\mathbf{r}}$ to construct $\tilde{\mathbf{r}}$ such that $c\tilde{\mathbf{r}}'Q = \mathbf{v}'$ and hence we would have

$$F(\mathbf{x}^*) - F(\mathbf{x}) \leq c\tilde{\mathbf{r}}'Q(\mathbf{x} - \mathbf{x}^*),$$

meaning $(\mathbf{x}^*, \tilde{\mathbf{r}})$ satisfies the saddle point inequality. This shows that $(\mathbf{x}^*, \tilde{\mathbf{r}})$ is an optimal primal-dual solution (see section 6 of [59]). Moreover, we choose $\tilde{\mathbf{r}}$ to satisfy the statement of lemma.

Let $Q = \sum_{i=1}^r \mathbf{u}_i\sigma_i\mathbf{v}_i'$ be the singular value decomposition of Q, where rank$(Q) = r$. Since $c\hat{\mathbf{r}}'Q = \mathbf{v}'$, $\mathbf{v}$ belongs to the span of $\{c\mathbf{v}_1, \ldots, c\mathbf{v}_r\}$ and can be written as $\mathbf{v} = c\sum_{i=1}^r c_i\mathbf{v}_i$ for some coefficients c_i's . Let $\tilde{\mathbf{r}} = \sum_{i=1}^r \frac{c_i}{\sigma_i}\mathbf{u}_i$. By this choice of $\tilde{\mathbf{r}}$ we have $c\tilde{\mathbf{r}}'Q = c\sum_{i=1}^r c_i\mathbf{v}_i = \mathbf{v}'$. This choice also yields

$$\begin{aligned}||\tilde{\mathbf{r}}||^2 &= \sum_{i=1}^r \frac{c_i^2}{\sigma_i^2} \leq \sum_{i=1}^r \frac{c_i^2}{\tilde{\lambda}_m} = \frac{1}{\sigma^2_{\text{min}}}\sum_{i=1}^r c_i^2 \\ &= \frac{1}{c^2\tilde{\lambda}_m}||\mathbf{v}||^2 \leq \frac{1}{c^2\tilde{\lambda}_m}U^2,\end{aligned}$$

where we used the bound on the subgradient to obtain the last inequality. Since $\tilde{\mathbf{r}}'Q = \mathbf{v}' \in \partial F(\mathbf{x}^*)$, $(\mathbf{x}^*, \tilde{\mathbf{r}})$ satisfies the saddle point inequality.

■

Next, we use this lemma to analyze the network effect. Using Theorem 1 with zero initial condition, we have

$$|F(\hat{\mathbf{x}}(T)) - F(\mathbf{x}^*)| \leq \frac{c}{2T}||\mathbf{x}^*||^2_{M-A'D^{-1}A} + \frac{2}{cT}\frac{U^2}{\tilde{\lambda}_m},$$

and

$$||Q\hat{\mathbf{x}}(T)|| \leq \frac{1}{2T}||\mathbf{x}^*||^2_{M-A'D^{-1}A} + \frac{1}{2T}\left(2 + 2\frac{U^2}{c^2\tilde{\lambda}_m}\right).$$

Using $||\mathbf{x}^*||^2_{M-A'D^{-1}A} \leq \lambda_M||\mathbf{x}^*||_2^2$ completes the proof.

F. Proof of Lemma 3

For any i, since $f_i(x) - \frac{\nu_{f_i}}{2}||x||^2$ is convex, $\nabla(f_i(x) - \frac{\nu_{f_i}}{2}||x||^2)$ is monotone and we have

$$\langle \nabla f_i(x) - \nabla f_i(y), x - y\rangle - \nu_{f_i}||x - y||^2 \geq 0.$$

Since $\nu \leq \nu_{f_i}$, we obtain

$$\langle \nabla f_i(x) - \nabla f_i(y), x - y\rangle - \nu||x - y||^2 \geq 0,$$

which results in

$$(\nabla F(\mathbf{x}) - \nabla F(\mathbf{y}))'(\mathbf{x} - \mathbf{y}) \geq \nu||\mathbf{x} - \mathbf{y}||_2^2,$$

this completes the proof of part (a). We now prove part (b). For any $x, y \in \mathbb{R}^n$, we have

$$\begin{aligned}f_i(y) &= f_i(x) + \int_0^1 \langle \nabla f_i(x + \tau(y - x)), y - x\rangle\, d\tau \\ &= f_i(x) + \langle \nabla f_i(x), y - x\rangle \\ &+ \int_0^1 \langle \nabla f_i((1-\tau)x + \tau y) - \nabla f_i(x), y - x\rangle\, d\tau \\ &\leq f_i(x) + \langle \nabla f(x), y - x\rangle \\ &+ \int_0^1 ||\nabla f_i((1-\tau)x + \tau y) - \nabla f_i(x)||||y - x||d\tau \\ &\leq f_i(x) + \langle \nabla f_i(x), y - x\rangle + L_{f_i}||y - x||^2\int_0^1 \tau d\tau \\ &= f_i(x) + \langle \nabla f_i(x), y - x\rangle + \frac{L_{f_i}}{2}||y - x||^2.\end{aligned}$$

Let $\phi_x(y) = f_i(y) - \langle \nabla f_i(x), y\rangle$. Note that $\phi_x(y)$ has Lipschitz gradient with parameter L_{f_i}. Moreover, we have that $\min_y \phi_x(y) = \phi_x(x)$, since

$$\nabla\phi_x(y) = \nabla f_i(y) - \nabla f_i(x)$$

is zero for $y = x$. Therefore, using the previous relation, we have that

$$\begin{aligned}\phi_x(y) - \phi_x(x) &= f_i(y) - f_i(x) - \langle \nabla f_i(x), y - x\rangle \\ &\geq \frac{1}{2L_{f_i}}||\nabla\phi_x(y)|| = \frac{1}{2L_{f_i}}||\nabla f_i(y) - \nabla f_i(x)||^2.\end{aligned}$$

Using the previous relation , we have

$$f_i(y) - f_i(x) - \langle \nabla f_i(x), y - x\rangle \geq \frac{1}{2L_{f_i}}||\nabla f_i(y) - \nabla f_i(x)||^2.$$

We also have

$$f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y\rangle \geq \frac{1}{2L_{f_i}}||\nabla f_i(y) - \nabla f_i(x)||^2.$$

We add the two preceding relations to obtain

$$\langle \nabla f_i(x) - \nabla f_i(y), x - y\rangle \geq \frac{1}{L_{f_i}}||\nabla f_i(y) - \nabla f_i(x)||^2,$$

for any $i = 1, \ldots, n$. Combining this relation for all i's completes the proof.

Finally, we prove part (c). Let $h = (h_1', \ldots, h_n')'$. By definition of subgradient, for any $i \in V$ we have

$$h_i(x_i(t+1))'(x_i - y_i) \geq f_i(x_i) - f_i(y_i).$$

Taking summation of this inequality for all $i = 1, \ldots, n$ shows that

$$h(\mathbf{x})'(\mathbf{x} - \mathbf{y}) \geq F(\mathbf{x}) - F(\mathbf{y}).$$

G. Proof of Theorem 3

We first show two lemmas that we use in the proof of this theorem. The first lemma shows that both matrices $M - A'D^{-1}A$ and $A'D^{-1}A$ are positive semidefinite and the second lemma shows a relation between the sequences $\mathbf{q}(t)$, $\mathbf{r}(t)$, and $\mathbf{x}(t)$ same as the one shown in Lemma 7.

Lemma 6. *The matrices $M - A'D^{-1}A$ and $A'D^{-1}A$ are positive semidefinite.*

Proof: Both matrices are clearly symmetric. We first show $M - A'D^{-1}A$ is positive semidefinite. By definition, we have

$$[M - A'D^{-1}A]_{ii} = M_{ii} - \sum_l A_{li}A_{li}\frac{1}{d_l+1} = \sum_l A_{li}^2\frac{d_l}{d_l+1}.$$

We also have

$$[M - A'D^{-1}A]_{ij} = -\sum_l A_{li}A_{lj}\frac{1}{d_l+1}.$$

Therefore,

$$\sum_{j\neq i} |[M - A'D^{-1}A]_{ij}\,| = \sum_{j\neq i} |\sum_l A_{li}A_{lj}\frac{1}{d_l+1}\,|$$
$$\leq \sum_l |A_{li}\,|\frac{1}{d_l+1}\,|\sum_{j\neq i} A_{lj}\,| = \sum_l A_{li}^2\frac{1}{d_l+1},$$

where we used the fact that $\sum_{j=1}^n A_{lj} = 0$, for any l. Therefore, by Greshgorin Circle Theorem, for any eigen value μ of $M - A'D^{-1}A$, for some i we have

$$|\mu - [M - A'D^{-1}A]_{ii}| \leq \sum_{j\neq i} |[M - A'D^{-1}A]_{ij}|,$$

which leads to

$$\mu \geq [M - A'D^{-1}A]_{ii} - \sum_{j\neq i} |[M - A'D^{-1}A]_{ij}\,|$$
$$\geq \sum_l A_{li}^2\left(\frac{d_l-1}{d_l+1}\right) \geq 0,$$

where we used the fact that $d_l \geq 1$ that evidently holds. We next show that $A'D^{-1}A$ is positive semidefinite. We have

$$[A'D^{-1}A]_{ii} = \sum_l A_{li}^2\frac{1}{d_l+1}$$

We also have

$$\sum_{j\neq i} |[A'D^{-1}A]_{ij}\,| = \sum_{j\neq i} |\sum_l A_{li}A_{lj}\frac{1}{d_l+1}\,|$$
$$\leq \sum_l |A_{li}\,|\frac{1}{d_l+1}\,|\sum_{j\neq i} A_{lj}\,| = \sum_l A_{li}^2\frac{1}{d_l+1}.$$

Since $[A'D^{-1}A]_{ii} \geq \sum_{j\neq i} |[A'D^{-1}A]_{ij}\,|$, similarly, by Greshgorin Circle Theorem, the matrix $A'D^{-1}A$ is positive semidefinite. ■

Lemma 7. *Suppose Assumptions 1 and 2 hold. For differentiable functions, the sequence $\{\mathbf{x}(t), \mathbf{r}(t)\}_{t=0}^{\infty}$ satisfies*

$$(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))$$
$$= -Q(\mathbf{r}(t+1) - \mathbf{r}^*) - \frac{1}{c}(\nabla F(\mathbf{x}(t+1)) - \nabla F(\mathbf{x}^*)),$$

for some $\mathbf{r}^*$ that satisfies $Q\mathbf{r}^* + \frac{1}{c}\nabla F(\mathbf{x}^*) = 0$. Moreover, $\mathbf{r}^*$ belongs to the column span of Q.

Proof: Using Lemma 2 for differentiable functions we have

$$M(\mathbf{x}(t+1) - \mathbf{x}(t)) = -\frac{1}{c}\nabla F(\mathbf{x}(t+1))$$
$$-(A'D^{-1}A)\mathbf{x}(t) - (A'D^{-1}A)\sum_{s=0}^{t}\mathbf{x}(s).$$

We subtract $(A'D^{-1}A)\mathbf{x}(t+1)$ from both sides and rearrange the terms to obtain

$$(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))$$
$$= -A'D^{-1}A\sum_{s=0}^{t+1}\mathbf{x}(s) - \frac{1}{c}\nabla F(\mathbf{x}(t+1)).$$

Using $QQ = A'D^{-1}A$, yields

$$(M - A'D^{-1}A)(\mathbf{x}(t+1) - \mathbf{x}(t))$$
$$= -Q\mathbf{r}(t+1) - \frac{1}{c}\nabla F(\mathbf{x}(t+1)).$$

We next show there exist $\mathbf{r}^*$ such that $Q\mathbf{r}^* + \frac{1}{c}\nabla F(\mathbf{x}^*) = 0$. First note that both column space (range) and null space of Q and $A'D^{-1}A$ are the same. Since $\text{span}(Q) \oplus \text{null}(Q) = \mathbb{R}^n$, we have $\nabla F(\mathbf{x}^*) \in \text{span}(Q) \oplus \text{null}(Q) = \text{span}(Q) \oplus \text{span}\{\mathbf{1}\}$ as $\text{null}(Q) = \text{span}\{\mathbf{1}\}$. Since $\mathbf{1}'\nabla F(\mathbf{x}^*) = 0$, we can write $\nabla F(\mathbf{x}^*)$ as a linear combination of column vectors of Q. Therefore, there exist $\mathbf{r}$ such that $\frac{1}{c}\nabla F(\mathbf{x}^*) = -Q\mathbf{r}$. Let $\mathbf{r}^* = \text{Proj}_Q\mathbf{r}$ to obtain $Q\mathbf{r} = Q\mathbf{r}^*$ where $\mathbf{r}^*$ lies in the column space of Q. Part (b) simply follows from the same lines of argument. ■

Back to the proof of Theorem 3: Note that since $M - A'D^{-1}A$ is positive semidefinite,

$$\langle .,. \rangle : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \mapsto \mathbb{R}$$
$$\langle \mathbf{q}_1, \mathbf{q}_2 \rangle = \mathbf{q}_1' G \mathbf{q}_2,$$

where

$$G = \begin{pmatrix} I & 0 \\ 0 & M - A'D^{-1}A \end{pmatrix}$$

is a semi-inner product.[2] We first show that for a δ given by the statement of theorem, we have

$$||\mathbf{q}(t+1) - \mathbf{q}^*||_G^2 \leq \left(\frac{1}{1+\delta}\right)||\mathbf{q}(t) - \mathbf{q}^*||_G^2. \quad (18)$$

[2] This means it satisfies conjugate symmetry, linearity and semipositive-definiteness (instead of positive-definiteness).

Using Lemma (3) and Lemma (7), we have

$$
\begin{aligned}
&\frac{2}{c}\nu||\mathbf{x}(t+1)-\mathbf{x}^*||_2^2 \\
&\leq \frac{2}{c}(\mathbf{x}(t+1)-\mathbf{x}^*)'(\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)) \\
&= 2(\mathbf{x}(t+1)-\mathbf{x}^*)'(Q(\mathbf{r}^*-\mathbf{r}(t+1))) \\
&+ 2(\mathbf{x}(t+1)-\mathbf{x}^*)'(M-A'D^{-1}A)(\mathbf{x}(t)-\mathbf{x}(t+1)) \\
&= 2(\mathbf{r}(t+1)-\mathbf{r}(t))'(\mathbf{r}^*-\mathbf{r}(t+1)) \\
&+ 2(\mathbf{x}(t+1)-\mathbf{x}(t))'(M-A'D^{-1}A)(\mathbf{x}^*-\mathbf{x}(t+1)) \\
&= 2(\mathbf{q}(t+1)-\mathbf{q}(t))'G(\mathbf{q}^*-\mathbf{r}(t+1)) \\
&= ||\mathbf{q}(t)-\mathbf{q}^*||_G^2-||\mathbf{q}(t+1)-\mathbf{q}^*||_G^2-||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2.
\end{aligned}
\tag{19}
$$

Again, using Lemma (3) and Lemma (7), we have

$$
\begin{aligned}
&\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2 \\
&\leq ||\mathbf{q}(t)-\mathbf{q}^*||_G^2-||\mathbf{q}(t+1)-\mathbf{q}^*||_G^2-||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2.
\end{aligned}
\tag{20}
$$

Using (19) and (20), for any $\beta \in (0,1)$, we have

$$
\begin{aligned}
&\beta\frac{2}{c}\nu||\mathbf{x}(t+1)-\mathbf{x}^*||_2^2 \\
&\quad + (1-\beta)\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2 \qquad (21)\\
&\leq ||\mathbf{q}(t)-\mathbf{q}^*||_G^2-||\mathbf{q}(t+1)-\mathbf{q}^*||_G^2-||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2. \qquad (22)
\end{aligned}
$$

This yields to

$$
\begin{aligned}
&||\mathbf{q}(t)-\mathbf{q}^*||_G^2-||\mathbf{q}(t+1)-\mathbf{q}^*||_G^2 \\
&\geq ||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2+\beta\frac{2}{c}\nu||\mathbf{x}(t+1)-\mathbf{x}^*||_2^2 \\
&+ (1-\beta)\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2.
\end{aligned}
\tag{23}
$$

Comparing this relation with (18), it remains to show

$$
\begin{aligned}
&||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2+\beta\frac{2}{c}\nu||\mathbf{x}(t+1)-\mathbf{x}^*||_2^2 \\
&+ (1-\beta)\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2 \\
&\geq \delta||\mathbf{q}(t+1)-\mathbf{q}^*||_G^2,
\end{aligned}
$$

which is equivalent to

$$
\begin{aligned}
&||\mathbf{q}(t)-\mathbf{q}(t+1)||_G^2+||\mathbf{x}(t+1)-\mathbf{x}^*||^2_{\frac{2\nu t}{c}I-\delta(M-A'D^{-1}A)} \\
&+ (1-\beta)\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2 \\
&\geq \delta||\mathbf{r}(t+1)-\mathbf{r}^*||_2^2.
\end{aligned}
\tag{24}
$$

Using Lemma (6), in order to show this inequality it suffices to show

$$
\begin{aligned}
&||\mathbf{x}(t+1)-\mathbf{x}^*||^2_{\frac{2\nu t}{c}I-\delta(M-A'D^{-1}A)} \\
&+ (1-\beta)\frac{2}{c}\frac{1}{L}||\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*)||_2^2 \\
&\geq \delta||\mathbf{r}(t+1)-\mathbf{r}^*||_2^2.
\end{aligned}
\tag{25}
$$

Since both $\mathbf{r}(t+1)$ and $\mathbf{r}^*$ are orthogonal to $\mathbf{1}$ and null$(Q) =$ span$(\{\mathbf{1}\})$, using Lemma (7), we obtain

$$
\begin{aligned}
&\delta||(\mathbf{r}(t+1)-\mathbf{r}^*)||_2^2 \leq \frac{\delta}{\tilde{\lambda}_m}||Q(\mathbf{r}(t+1)-\mathbf{r}^*)||_2^2 \\
&\leq \frac{\delta}{\tilde{\lambda}_m}||(M-A'D^{-1}A)(\mathbf{x}(t+1)-\mathbf{x}^*) \\
&- \frac{1}{c}(\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*))||_2^2 \\
&\leq \frac{2\delta}{\tilde{\lambda}_m}||(M-A'D^{-1}A)(\mathbf{x}(t+1)-\mathbf{x}^*)||_2^2 \\
&+ \frac{2\delta}{\tilde{\lambda}_m}||(\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*))||_2^2 \\
&\leq \frac{2\delta\lambda_M}{\tilde{\lambda}_m}||\mathbf{x}(t+1)-\mathbf{x}^*||^2_{M-A'D^{-1}A} \\
&+ \frac{2\delta}{\tilde{\lambda}_m}||(\nabla F(\mathbf{x}(t+1))-\nabla F(\mathbf{x}^*))||_2^2
\end{aligned}
\tag{26}
$$

Comparing (26) and (25), it suffices to have

$$
\delta \leq \min\left\{\frac{2\beta\nu}{c\lambda_M(1+\frac{2}{\tilde{\lambda}_m})}, \frac{(1-\beta)c\tilde{\lambda}_m}{L}\right\}.
$$

This shows that (18) holds. Using (18) along with (19) completes the proof.

H. Proof of Theorem 4

The largest possible δ that satisfies the constraint given in Theorem 1 by maximizing over $\beta \in (0,1)$ is the solution of

$$
\frac{2\beta\nu}{c\lambda_M(1+\frac{2}{\tilde{\lambda}_m})} = \frac{(1-\beta)c\tilde{\lambda}_m}{L},
\tag{27}
$$

which is $\beta^* = \frac{c^2\lambda_M(2+\tilde{\lambda}_m)}{2\nu L+c^2\lambda_M(2+\tilde{\lambda}_m)}$. This in turn shows that the maximum δ is equal to $\delta = \frac{2\beta^*\nu}{c\lambda_M(1+\frac{2}{\tilde{\lambda}_m})}$. We now maximize δ over choices of c, leading to

$$
\delta^* = \frac{1}{2}\sqrt{\frac{2\tilde{\lambda}_m^2}{\lambda_M(2+\tilde{\lambda_m})}\frac{1}{\kappa_f}}.
$$

I. Proof of Proposition 3

The bound provided in Theorem 2 depends on $\tilde{\lambda}_m$. We have that

$$
\tilde{\lambda}_m \geq \frac{1}{d_{\text{max}}+1}a(G)^2,
$$

where $a(G)$ is the algebraic connectivity of the graph which is the smallest non-zero eigenvalue of the Laplacian matrix. Moreover, we have that

$$
||M-A'D^{-1}A||_2 \leq d_{\text{max}}(d_{\text{max+1}}) + \frac{4d_{\text{max}}^2}{d_{\text{min}}+1}.
$$

Plugging these two bounds in Theorem 2 an using the fact that $d_{\text{min}} \geq 1$ completes the proof.

J. Proof of Proposition 4

Using Theorem 4, for large enough κ_f(small enough δ^*) we have $\frac{1}{\log(1+\delta^*)} \geq \frac{1}{\delta^*}$, in order to have $||\mathbf{x}(t) - \mathbf{x}^*||_2 \leq \epsilon$, we need to have

$$t \geq \frac{1}{\log\frac{1}{\rho}} \left(2\log(\frac{1}{\epsilon}) - \log(\frac{c^*}{2\nu}||\mathbf{q}(0) - \mathbf{q}^*||_G^2) \right)$$
$$\geq \sqrt{\kappa_f}\frac{2\sqrt{\lambda_M(2+\tilde{\lambda}_m)}}{\sqrt{2}\tilde{\lambda}_m} \left(2\log(\frac{1}{\epsilon}) - \log(\frac{c^*}{2\nu}||\mathbf{q}(0) - \mathbf{q}^*||_G^2) \right). \tag{28}$$

This shows that $O\left(\sqrt{\kappa_f}\frac{\sqrt{\lambda_M(2+\tilde{\lambda}_m)}}{\tilde{\lambda}_m}\log(\frac{1}{\epsilon})\right)$ iterations suffice to have $||\mathbf{x}(t) - \mathbf{x}^*||_2 \leq \epsilon$. This bound depends on $\tilde{\lambda}_m$ and λ_M. We have the following bounds

$$\frac{1}{d_{\text{min}}+1}a(G)^2 \geq \tilde{\lambda}_m \geq \frac{1}{d_{\text{max}}+1}a(G)^2.$$

and

$$\lambda_M \leq d_{\text{max}}(d_{\text{max}+1}) + \frac{\lambda_{\text{max}}(A)^2}{d_{\text{min}}+1}.$$

Using these two bounds along with $\lambda_{\text{max}}(A) \leq 2d_{\text{max}}$, we obtain

$$\frac{\lambda_M(2+\tilde{\lambda}_m)}{\tilde{\lambda}_m^2} \leq 16\frac{d_{\text{max}}^4}{d_{\text{min}}a^2(G)}.$$

Plugging this bound into (28) completes the proof.

References

[1] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, 2007.

[2] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *The Journal of Machine Learning Research, 2012*.

[3] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems, 2011*.

[4] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems, 2011*.

[5] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and Y. Zhang, "Optimality guarantees for distributed statistical estimation," *arXiv preprint arXiv:1405.0782*, 2014.

[6] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing, 2014*.

[7] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *International symposium on Information processing in sensor networks, 2004*.

[8] S. H. Low and D. E. Lapsley, "Optimization flow control?i: basic algorithm and convergence," *IEEE/ACM Transactions on Networking (TON), 1999*.

[9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, 2003.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, 1995.

[11] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Advances in Neural Information Processing Systems*, 2013.

[12] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression," in *Conference on Learning Theory*, 2013.

[13] I. Mitliagkas, C. Caramanis, and P. Jain, "Memory limited, streaming pca," in *Advances in Neural Information Processing Systems*, 2013.

[14] Y. Zhang, M. J. Wainwright, and J. C. Duchi, "Communication-efficient algorithms for statistical optimization," in *Advances in Neural Information Processing Systems*, 2012.

[15] Y. Zhang and L. Xiao, "Communication-efficient distributed optimization of self-concordant empirical loss," *arXiv preprint arXiv:1501.00263*, 2015.

[16] M. E. Hellman and T. M. Cover, "Learning with finite memory," *The Annals of Mathematical Statistics*, 1970.

[17] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *The Journal of Machine Learning Research*, 2010.

[18] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Advances in Neural Information Processing Systems*, 2014.

[19] H. Wang, A. Banerjee, C.-J. Hsieh, P. K. Ravikumar, and I. S. Dhillon, "Large scale distributed sparse precision estimation," in *Advances in Neural Information Processing Systems*, 2013.

[20] Ö. Aslan, H. Cheng, X. Zhang, and D. Schuurmans, "Convex two-layer modeling," in *Advances in Neural Information Processing Systems*, 2013.

[21] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Advances in Neural Information Processing Systems*, 2013.

[22] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." DTIC Document, Tech. Rep., 1984.

[23] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control, 1986*.

[24] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, 2010.

[25] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications, 2010*.

[26] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing, 2012*.

[27] D. Jakovetic, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control, 2014*.

[28] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *arXiv preprint arXiv:1404.6264*, 2014.

[29] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, 2009.

[30] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, 2012.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[32] J. Eckstein, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," *RUTCOR Research Reports*, vol. 32, 2012.

[33] R. Glowinski and A. Marroco, "Sur l'approximation, par e?le?ments finis d'ordre un, et la re?solution, par pe?nalisation-dualite? d'une classe de proble?mes de dirichlet non line?aires," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 1975.

[34] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, 1976.

[35] D. P. Bertsekas and J. Eckstein, "Dual coordinate step methods for linear network flow problems," *Mathematical Programming*, 1988.

[36] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An admm algorithm for a class of total variation regularized estimation problems," in *Preprints of the 16th IFAC Symposium on System Identification*, 2012.

[37] H. Sedghi, A. Anandkumar, and E. Jonckheere, "Multi-step stochastic admm in high dimensions: Applications to sparse optimization and matrix decomposition," in *Advances in Neural Information Processing Systems*, 2014.

[38] H. Wang and A. Banerjee, "Online alternating direction method," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012.

[39] C. Zhang, H. Lee, and K. G. Shin, "Efficient distributed linear classification algorithms via the alternating direction method of multipliers," in *International Conference on Artificial Intelligence and Statistics*, 2012.

[40] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014.

[41] J. F. Mota, J. Xavier, P. M. Aguiar, and M. Puschel, "Basis pursuit in sensor networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2916–2919.

[42] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy linkspart i: Distributed estimation of deterministic signals," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 350–364, 2008.

[43] N. S. Aybat and G. Iyengar, "An alternating direction method with increasing penalty for stable principal component pursuit," *Computational Optimization and Applications*, pp. 1–34, 2014.

[44] N. S. Aybat, S. Zarmehri, and S. Kumara, "An admm algorithm for clustering partially observed networks," *arXiv preprint arXiv:1410.3898*, 2014.

[45] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, 2014.

[46] S. Shtern, E. Wei, and A. Ozdaglar, "Distributed alternating direction method of multipliers (admm): Performance and network effects," in *Working paper*.

[47] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," DTIC Document, Tech. Rep., 2012.

[48] B. He and X. Yuan, "On the o(1/n) convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, 2012.

[49] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, 1979.

[50] J. Eckstein and M. C. Ferris, "Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control," *INFORMS Journal on Computing*, 1998.

[51] P. Patrinos, L. Stella, and A. Bemporad, "Douglas-rachford splitting: complexity estimates and accelerated variants," *arXiv preprint arXiv:1407.6723*, 2014.

[52] ——, "Forward-backward truncated newton methods for large-scale convex composite optimization," *arXiv preprint arXiv:1402.6655*, 2014.

[53] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," *arXiv preprint arXiv:1406.4834*, 2014.

[54] ——, "Convergence rates of relaxed peaceman-rachford and admm under regularity assumptions," *arXiv preprint arXiv:1407.5210*, 2014.

[55] P. Giselsson and S. Boyd, "Diagonal scaling in douglas-rachford splitting and admm," in *IEEE Conference on Decision and Control*, 2014.

[56] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of admm," *arXiv preprint arXiv:1502.02009*, 2015.

[57] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *arXiv preprint arXiv:1408.3595*, 2014.

[58] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.

[59] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex analysis and optimization*. Athena Scientific Belmont, 2003.

[60] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997.

[61] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, 1973.

[62] N. M. M. de Abreu, "Old and new results on algebraic connectivity of graphs," *Linear algebra and its applications*, 2007.