Projected Gradient Method for Decentralized Optimization over Time-Varying Networks

Alexander Rogozin * Alexander Gasnikov **

* Moscow Institute of Physics and Technology, Dolgoprudny, Russia
(e-mail: aleksandr.rogozin@phytech.edu).
** Moscow Institute of Physics and Technology, Dolgoprudny, Russia
Institute for Information Transmission Problems, Moscow, Russia,
Higher school of economics, Moscow, Russia
(e-mail: gasnikov@yandex.ru).

Abstract: Decentralized distributed optimization over time-varying graphs (networks) is nowadays a very popular branch of research in optimization theory and consensus theory. One of the motivations to consider such networks is an application to drone networks. However, the first theoretical results in this branch appeared only five years ago (Nedić and Olshevsky (2014)). The first results about the possibility of geometric rates of convergence for strongly convex smooth optimization problems on such networks were obtained only two years ago in Nedić et al. (2017). In this paper, we improve the rate of geometric convergence in Nedić et al. (2017) for the considered class of problems, using an original penalty method trick and robustness of projected gradient descent.

Keywords: Convex optimization, distributed optimization

1. INTRODUCTION

The theory of decentralized distributed optimization goes back to Bertsekas and Tsitsiklis (1989). In the last few years this branch of research has aroused great interest in optimization community. A set of papers proposing optimal algorithms for convex optimization problems of sum-type has appeared. See for example Arjevani and Shamir (2015); Scaman et al. (2017); Lan et al. (2018); Dvinskikh and Gasnikov (2019) and references therein. In all these papers, authors consider sum-type convex target functions and aim at proposing algorithms that find the solution with required accuracy and make the best possible number of communications steps and number of oracle calls (gradient calculations of terms in the sum). In Dvinskikh and Gasnikov (2019), it is mentioned that the theory of optimal decentralized distributed algorithms looks very close to the analogous theory for ordinary convex optimization (Nemirovskii and Yudin (1983); Nesterov (2013); Bubeck (2015)). Roughly speaking, in a first approximation, decentralized distributed optimization comes down to the theory of optimal methods and this theory is significantly based on the theory of non-distributed optimal methods.

In decentralized distributed optimization over time-varying graphs, another situation takes place. The communication network topology changes from time to time, which can be caused by technical malfunctions such as loss of connection between the agents. Due to the many applications, the interest to these class of problems has grown significantly during the last few years. There appears a number of papers with theoretical analysis of rate of convergence for convex type problems: Nedić and Olshevsky (2014); Nedić et al. (2017); Maros and Jaldén (2018); Lü et al. (2018); Van Scoy and Lessard (2019). But there is still a big gap between the theory for decentralized optimization on fixed graphs and the theory over time-varying graphs. The attempt to close this gap (specifically, to develop optimal methods) for the moment required very restricted additional conditions (Rogozin et al. (2019)).

In this paper, we make a step in the direction of development of optimal methods over time-varying graphs: we propose non-accelerated gradient descent for smooth strongly convex target functions of sum-type. Our analysis is based on reformulation of initial optimization problem as convex optimization problem under affine constraints. These constraints change from time to time but still determine the same hyperplane. Then we use projected gradient descent. In order to solve auxiliary problem (to find a projection on hyperplane) we use non accelerated consensus type algorithms (see Hendrikx et al. (2018) and references therein for comparison) that can also be interpreted as gradient descent for special penalized optimization problem (Gasnikov (2017); Dvinskikh and Gasnikov (2019)). Note that proposed analysis of external non-accelerated gradient descent method can be generalized for the case of accelerated gradient method. We plan to do it in subsequent works.

This paper is organized as follows. In Section 2, we recall some basic definitions and show how to reformulate a prob-
lem with affine constraints by introducing a penalty. In Section 3, we analyse the performance of gradient descent on a time-varying function. We introduce decentralized projected gradient method in Section 4 and analyze its convergence using results of Section 3. Finally, we provide numerical experiments and comparison to other methods in Section 5.

2. PRELIMINARIES

2.1 Strong Convexity and Smoothness

Strongly convex and smooth functions are the focus of this paper.

Definition 1. Let \( \mathbb{X} \) be either \( \mathbb{R}^d \) with 2-norm or \( \mathbb{R}^{d \times n} \) with Frobenius norm. A differentiable function \( f : \mathbb{X} \to \mathbb{R} \) is called

- **convex**, if for any \( x, y \in \mathbb{X} \)
  \[ f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle; \]
- **\( \mu \)-strongly convex**, if for any \( x, y \in \mathbb{X} \)
  \[ f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2; \]
- **L-smooth**, if its gradient \( \nabla f(x) \) is L-Lipschitz, i.e. for any \( x, y \in \mathbb{X} \)
  \[ \| \nabla f(y) - \nabla f(x) \| \leq L \| y - x \|, \]
  or, equivalently, for all \( x, y \in \mathbb{X} \)
  \[ f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \| y - x \|^2. \]

2.2 Graph Laplacian

In this paper, a communication network is represented by a connected undirected graph \( \mathcal{G} = (V, E) \).

Definition 2. For an undirected graph \( \mathcal{G} = (V, E) \) with \( |V| = n \) nodes, its **Laplacian** is a matrix \( W \in \mathbb{R}^{n \times n} \) such that

\[
[W]_{ij} = \begin{cases} 
\deg i, & i = j \\
-1, & (i, j) \in E \\
0, & \text{else}
\end{cases}
\]

(1)

In the statement below, we list the basic Laplacian properties, which can be obtained using Perron-Frobenius theorem (Nikaido (1968)).

Proposition 3.

- \( W \) is positive semidefinite;
- If graph \( \mathcal{G} \) is connected, then \( \forall x \in \mathbb{R}^n \quad \text{Ker } W = \text{Span}(1) \). Moreover, \( \text{Ker } \sqrt{W} = \text{Ker } W = \text{Span}(1) \).

2.3 Convex Problem with Affine Constraints

As will be shown later in the paper, decentralized optimization problems may be reformulated as problems with affine constraints. Consider optimization problem

\[ f(x) \to \min_{A x = 0} \] (2)

for some convex function \( f : \mathbb{R}^d \to \mathbb{R} \) and symmetric positive semidefinite matrix \( A \in \mathbb{R}^{d \times d} \). Let \( y^* \) be the solution of Lagrange dual to (2) with minimal norm and \( R_y = \| y^* \| \). Introduce a penalized problem

\[ f_A(x) = f(x) + \frac{R_y^2}{\varepsilon} \| A x \|^2 \to \min_{x \in \mathbb{R}^d} \]

The quantity \( R_y \) can be bounded as (Lan et al. (2018))

\[ R_y^2 \leq \frac{\| \nabla f(x^*) \|^2}{\lambda^+_\min(A^2)}, \]

where \( x^* \) is the solution of (2) and \( \lambda^+_\min(\cdot) \) denotes the minimum positive eigenvalue of the corresponding matrix.

Proposition 4. Let \( x \in \mathbb{R}^d \) and \( f_A(x) = \min_{A x = 0} f_A(x) < \varepsilon \).

Then

\[
\begin{cases} 
 f(x_N) - \min_{A x = 0} f(x) < \varepsilon \\
 \| A x_N \| < 2\varepsilon/R_y
\end{cases}
\]

Proof. See Dvinskikh and Gasnikov (2019); Gasnikov (2018), Theorem 1 in ? .

Proposition 5. Denote \( x^* = \arg \min_{A x = 0} f(x) \) and \( x^*_\lambda = \arg \min f_A(x) \).

Then \( \| x^*_\lambda - x^* \| \leq \sqrt{\varepsilon a} \), where \( a \) is defined as

\[ a = \frac{4\| \nabla f(x^*) \|}{\mu R_y \lambda^+_\min(A)} \] (3)

Proof. Using Proposition 4, since \( f(x^*_\lambda) - f(x^*) < \varepsilon \), it holds \( \| A x^*_\lambda \| = \| A x^* \| < 2\varepsilon/R_y \). Denote \( \Delta x = x^*_\lambda - x^* = \Delta x_\perp + \Delta x_\parallel \), where \( \Delta x_\parallel = \Pi_{\text{Ker } A}(\Delta x) \).

1. First, we estimate \( \| x_\parallel \| \).

\[ \lambda^+_\min(A) \| \Delta x_\parallel \| \leq \| A \Delta x_\parallel \| = \| A x^*_\lambda \| < 2\varepsilon/R_y \]

\[ \| \Delta x_\parallel \| \leq \frac{2\varepsilon}{R_y \lambda^+_\min(A)} \]

2. Second, let us estimate \( \| x_\perp \| \). By strong convexity of \( f \):

\[
f(x^*_\lambda) \geq f(x^*) + \langle \nabla f(x^*), \Delta x_\perp \rangle + \frac{\mu}{2} \| \Delta x_\perp \|^2 \]

\[
\geq f(x^*) - \| \nabla f(x^*) \|_2 \| \Delta x_\perp \| + \frac{\mu}{2} \| \Delta x_\parallel \|^2 \]

\[
f(x^*_\lambda) - f(x^*) \geq \| \nabla f(x^*) \| \cdot \| \Delta x_\perp \| + \frac{\mu}{2} \| \Delta x_\parallel \|^2 + \frac{R_y^2}{\varepsilon} \| A x^*_\lambda \|^2 \]

\[
\| \Delta x_\perp \| \geq \sqrt{\varepsilon a} \]

3. GRADIENT DESCENT ON A TIME-VARYING FUNCTION

Definition 6. We call a series of functions \( g = \{ g_k \}_{k=1}^\infty \), \( g_k : \mathbb{R}^n \to \mathbb{R} \), a **time-varying function**. If each of
$g_k$ is convex/$\mu$-strongly convex/L-smooth, we call $g$ a convex/$\mu$-strongly convex/L-smooth time-varying function.

We define gradient descent on a time-varying function as

$$x_{k+1} = x_k - \gamma \nabla g_k(x_k)$$  \hspace{1cm} (4)

We are interested in convergence of the above method. In order to establish the rate, we need additional assumptions.

**Assumption 7.**

1. Time-varying function $g$ is $\mu$-strongly convex and $L$-smooth;
2. There exists $x^* \in \mathbb{R}^n$, $\varepsilon > 0$ such that for $k = 1, 2, \ldots$ it holds $\|\arg \min g_k(x) - x^*\|^2 \leq \varepsilon$.

Assumption 7 becomes realistic when time-varying function $g$ represents a functional with changing penalty. It is discussed later in the paper. First, we formulate preliminary facts that will be needed in analysis.

**Proposition 8.** For $\mu$-strongly convex $L$-smooth function $f$, it holds

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{\mu L}{\mu + L} \|x - y\|^2 + \frac{1}{\mu + L} \|\nabla f(x) - \nabla f(y)\|^2$$

**Proof.** See Theorem 2.1.11 in Nesterov (2013).

**Proposition 9.** Let $u, v$ be vectors of $\mathbb{R}^n$ of matrices of $\mathbb{R}^{d \times n}$ and $p$ be a positive scalar constant. Then

1. $\langle u, v \rangle \leq \frac{\|u\|^2}{2p} + \frac{p\|v\|^2}{2}$  \hspace{1cm} (5)

2. If $p < 1$, then

$$\|v\|^2 \geq p \|u\|^2 - \frac{p}{1 - p} \|v - u\|^2$$  \hspace{1cm} (6)

Here, if $u, v \in \mathbb{R}^n$, $\| \cdot \|$ denotes the 2-norm in $\mathbb{R}^n$, and if $u, v \in \mathbb{R}^{d \times n}$, $\| \cdot \|$ denotes Frobenius norm.

**Proof.**

1. Multiplying both sides by $2c$ yields $\|u - pv\|^2 \geq 0$.

2. Analogously, multiplying both sides by $1 - p$ leads to

$$(1 - p)\|v\|^2 \geq (p - p^2)\|u\|^2 - p\|v\|^2 + \|u\|^2 - 2\langle u, v \rangle$$

$$\|v\|^2 \geq -p^2\|u\|^2 + 2p\langle u, v \rangle$$

$$\|v - pu\|^2 \geq 0$$

Denote $r_k = \|x_k - x^*\|, x_k^* = \arg \min g_k(x), \Delta x_k = x_k^* - x^*$.

$$r_{k+1}^2 = \|x_k - x^* - \gamma \nabla g_k(x_k)\|^2$$

$$= r_k^2 + 2\gamma \|\nabla g_k(x_k)\| - 2\gamma \langle \nabla g_k(x_k), x_k - x^* \rangle$$

$$= r_k^2 + 2\gamma \|\nabla g_k(x_k)\| - 2\gamma \langle \nabla g_k(x_k) - \nabla g_k(x_k^*), x_k - x_k^* \rangle$$

$$\leq r_k^2 + 2\gamma \|\nabla g_k(x_k)\| - 2\gamma \|\nabla g_k(x_k) - \nabla g_k(x_k^*)\|$$

$$\leq r_k^2 + 2\gamma \|\nabla g_k(x_k)\|^2 - 2\gamma (\langle \nabla g_k(x_k), x_k^* - x^* \rangle)$$

$$\leq r_k^2 + 2\gamma \|\nabla g_k(x_k)\|^2 - 2\gamma (\|\nabla g_k(x_k)\|^2 - \|\nabla g_k(x_k) - \nabla g_k(x_k^*)\|^2)$$

$$\leq r_k^2 + 2\gamma \|\nabla g_k(x_k)\|^2 - 2\gamma (\|\nabla g_k(x_k)\|^2 - \|\nabla g_k(x_k) - \nabla g_k(x_k^*\|)^2)$$

$$\leq r_k^2 + 2\gamma \|\nabla g_k(x_k)\|^2 - 2\gamma (\|\nabla g_k(x_k) - \nabla g_k(x_k^*\|^2)$$

Now let us employ Proposition 9 for the under-braced terms.

1. Using (5) with $p = \mu + L$:

$$-2\gamma \langle \nabla g_k(x_k), x_k^* - x^* \rangle$$

$$\leq 2\gamma \|\nabla g_k(x_k)\|^2 (2(\mu + L) + 2\gamma \cdot \frac{\mu + L}{2} \|\Delta x_k\|^2$$

$$= \gamma \|\nabla g_k(x_k)\|^2 + (\mu + L) \|\Delta x_k\|^2$$

2. Using (6) with $p = 1/2, u = x_k - x^*, v = x_k - x_k^*$:

$$\|x_k - x_k^*\|^2 \geq \frac{1}{2} \|x_k - x^*\|^2 - \|x^* - x_k^*\|^2$$

Returning to (7) we obtain

$$r_{k+1}^2 \leq r_k^2 \left(1 - 2\gamma \frac{\mu L}{\mu + L} + \frac{1}{2}\right)$$

$$+ \frac{\mu L}{\mu + L} \|\Delta x_k\|^2 \left(\gamma - \frac{2\gamma}{\mu + L} + \frac{\gamma}{\mu + L}\right)$$

$$= r_k^2 \left(1 - \gamma \frac{\mu L}{\mu + L} + \|\nabla g_k(x_k)\|^2 \cdot \gamma \left(\gamma - \frac{1}{\mu + L}\right) \right)$$

$$+ \|\Delta x_k\|^2 \cdot \gamma \left(\frac{2\mu L}{\mu + L} + (\mu + L)\right)$$

Recall that here $\gamma$ is stepsize from (4). Setting $\gamma \in \left(0, \frac{1}{\mu + L}\right)$ leads to

$$r_{k+1}^2 \leq r_k^2 \left(1 - \gamma \frac{\mu L}{\mu + L}\right) + \|\Delta x_k\|^2 \cdot \gamma \left(\frac{2\mu L}{\mu + L} + (\mu + L)\right)$$

$$\leq r_k^2 \left(1 - \gamma \frac{\mu L}{\mu + L}\right) + \gamma \left(\frac{2\mu L}{\mu + L} + (\mu + L)\right) \cdot \varepsilon$$

$$\leq r_k^2 \left(1 - \gamma \frac{\mu L}{\mu + L}\right) + \gamma \cdot 3L_\varepsilon.$$  \hspace{1cm} (8)

In order to obtain linear convergence, let us formulate the following

**Lemma 10.** Let the following inequality hold:

$$r_k^2 \geq b\varepsilon$$  \hspace{1cm} (9)

where

$$b = 12 \frac{L}{\mu}$$  \hspace{1cm} (10)
Then
\[ r^2_{k+1} \leq r^2_k \left( 1 - \frac{\mu}{2} \right) \]

**Proof.** Consider (8) and rewrite it the following way:
\[ r^2_{k+1} \leq r^2_k \left( 1 - \frac{\mu}{2} \right) + 3\gamma L \varepsilon \]
The latter term is non-positive due to (9), (10), and the desired result follows.

Finally, we are ready to state a convergence result in terms of number of iterations.

**Theorem 11.** Under Assumption 7, after \( N \) steps of gradient descent with stepsize \( \gamma = \frac{1}{\mu + L} \), where \( N = O \left( \frac{1}{\mu} \log \frac{2}{\varepsilon} \right) \) and \( b \) is defined in (10), the following inequality holds:
\[ r^2_N \leq b \varepsilon \]

**Proof.** If \( r_k \geq b \varepsilon \), Lemma 10 works. If \( r_k < b \varepsilon \), then
\[ r^2_{k+1} \leq r^2_k \left( 1 - \frac{\mu}{2} \right) + 3\gamma L \varepsilon \]
\[ \leq 12 \frac{L}{\mu} \left( 1 - \frac{\mu}{2} \right) \varepsilon + 3\gamma L \varepsilon \leq 12 \frac{L}{\mu} \varepsilon = b \varepsilon. \] (11)
This means that once method achieves \( b \varepsilon \) accuracy after some \( N \) steps, its trajectory remains in \( b \varepsilon \)-region of \( x^* \), i.e. \( r^2_k \leq b \varepsilon \) for all \( k \geq N \). By Lemma 10, it is sufficient to make \( N = O \left( \frac{1}{\mu} \log \frac{2}{\varepsilon} \right) \) in order to obtain \( r^2_N \leq b \varepsilon \).

4. DECENTRALIZED PROJECTED GRADIENT METHOD

4.1 Problem Reformulation and Assumptions

Consider minimization of sum of convex functions:
\[ f(x) = \sum_{i=1}^{n} f_i(x) \rightarrow \min_{x \in \mathbb{R}^d} \]
(12)
We assume that every \( f_i \) is \( \mu_i \)-strongly convex and \( L_i \)-smooth. We seek to solve problem (12) in a decentralized setup, so that every node locally holds \( f_i \) and may exchange data with its neighbors. Moreover, we are interested in the time-varying case. This means the communication network changes with time and is represented by a sequence of graphs \( \{G_k\}_{k=1}^{\infty} \). Our analysis is restricted to the following

**Assumption 12.** Each of graphs \( \{G_k\}_{k=1}^{\infty} \) is connected.

Moreover, we introduce bounds on the graph Laplacian spectrum.

**Definition 13.** For each graph \( G_k \), let \( W_k \) be its Laplacian. Denote
\[ \theta_{\max} = \max_k \lambda_{\max}(W_k) \]
(13a)
\[ \theta_{\min} = \min_k \lambda_{\min}^+(W_k) \]
(13b)
\[ \chi = \frac{\theta_{\max}}{\theta_{\min}} \]
(13c)
Let us reformulate problem (12) in a following way.
\[ F(X) = \sum_{i=1}^{n} f_i(x_i) \rightarrow \min_{x_1, \ldots, x_n}. \] (14)
Here \( X \in \mathbb{R}^{d \times n} \) is a matrix consisting of columns \( x_1, \ldots, x_n \). The above representation means local copies \( x_i \) of parameter vector \( x \) are distributed over the agents in the network. Now, if every node computes \( \nabla f_i(x_i) \), then the gradient \( \nabla F(X) = [\nabla f_1(x_1), \ldots, \nabla f_n(x_n)] \) will be distributed all over the network. We will use notation \( \nabla F(X) \) in the analysis, although \( \nabla F(X) \) is not stored at one computational entity.

We call \( K \) a linear subspace in \( \mathbb{R}^{d \times n} \) determined by the constraint \( x_1 = \ldots = x_n \). Note that \( F \) defined in (14) is \( \mu_{\min} \)-strongly convex and \( L_{\max} \)-smooth on \( \mathbb{R}^{d \times n} \), but \( \mu_f/L \)-strongly convex and \( (L_f/n) \)-smooth on \( K \), where
\[ \mu_{\min} = \min_i \mu_i, \quad L_{\max} = \max_i L_i, \]
and \( \mu_f \) and \( L_f \) are strong convexity and smoothness constants of \( f \). Indeed, note that for any \( X, Y \in K \) it holds \( X = (x, \ldots, x), Y = (y, \ldots, y) \) and therefore
\[ F(X) = \sum_{i=1}^{n} f_i(x) = f(x), \quad F(Y) = f(y), \]
\[ \|Y - X\|^2 = n\|y - x\|^2 \]
\[ \langle \nabla F(X), Y - X \rangle = \sum_{i=1}^{n} \langle \nabla f_i(x), y - x \rangle = \langle \nabla f(x), y - x \rangle \]
\[ \langle \nabla F(Y), X - Y \rangle = \sum_{i=1}^{n} \langle \nabla f_i(x), y - x \rangle \]
\[ F(Y) \leq F(X) + \langle \nabla F(X), Y - X \rangle + \frac{\mu_f}{2n} \|Y - X\|^2 \]
\[ F(Y) \leq F(X) + \langle \nabla F(X), Y - X \rangle + \frac{L_f}{2n} \|Y - X\|^2 \]

4.2 Gradient Descent with Exact Projections

Let us consider a projected gradient method applied to problem (14).
\[ \Pi_{k+1} = \Pi_k - \gamma \text{Proj}_K(\nabla F(\Pi_k)) \] (16)
Choosing \( \Pi_0 \in K \) makes the method trajectory stay in \( K \), since \( K \) is a linear subspace. Therefore, the algorithm may be interpreted as a simple gradient descent on \( K \). Since function \( F \) defined in (14) is \( (\mu_f/n) \)-strongly convex and \( (L_f/n) \)-smooth on \( K \), the algorithm (16) requires \( O(L_f/n \log(1/\varepsilon)) \) iterations to achieve \( \varepsilon \)-solution of problem (14).

However, exact projected method cannot be run in a decentralized manner. In the next section, we introduce an inexact version of this algorithm and analyse its convergence.

4.3 Inexact Projected Gradient Descent

**Algorithm 1** Decentralized Projected GD

**Require:** Each node holds \( f_i(\cdot) \) and iteration number \( N \).
1: Initialize \( X_0 = [x_0, \ldots, x_0] \), choose \( c > 0 \).
2: for \( k = 0, 1, 2, \ldots, N - 1 \) do
3: \[ Y_{k+1} = X_k - \gamma \nabla F(X_k) \]
4: \[ X_{k+1} \approx \text{Proj}_K(Y_{k+1}) \]
5: \[ X_{k+1} \approx \text{Proj}_K(Y_{k+1}) \quad \|X_{k+1} - \text{Proj}_K(Y_{k+1})\|^2 \leq \varepsilon_1 \text{ and } X_{k+1} - \text{Proj}_K(Y_{k+1}) \in K \]
end for

Performing step 4 in a decentralized way on a time-varying graph is done by non-accelerated gradient descent and is discussed in later sections. Here we present a convergence result for Algorithm 1.
Theorem 14. After $N = O \left( \frac{L \log \left( \frac{L}{\epsilon} \right)}{\epsilon^2} \right)$ iterations, Algorithm 1 with $\epsilon_1 = \frac{\mu^2}{\gamma_0 \epsilon_1 \hat{L}_{max}}$ yields $X_N$ such that

$$
||X_N - X^*||^2 \leq \epsilon
$$

The proof of Theorem 14 is performed in Appendix A.

4.4 Finding Inexact Projection

In this section, we provide an algorithm for finding approximate value of $\text{Proj}_K(Y)$. We formulate projection as an optimization problem

$$
\frac{1}{2} ||X - Y||^2 \rightarrow \min_{X \in K}
$$

Suppose we are given a static connected graph $\mathcal{G}$. Using the fact that $\text{Ker} \mathcal{W} = K$, (see Proposition 3), the problem above can be rewritten as

$$
\frac{1}{2} ||X - Y||^2 \rightarrow \min_{X \in \mathcal{W}}
$$

Moreover, we can penalize the constraint $X \in \mathcal{W} = 0$ (see Proposition 4):

$$
\frac{1}{2} ||X - Y||^2 + \frac{R^2}{\epsilon_2} ||X \mathcal{W}||^2 \rightarrow \min_{X \in \mathbb{R}^{d \times n}}
$$

with some $R^2 \geq ||X - Y||^2$. However, communication graph changes with time and hence graph Laplacian $\mathcal{W}$ changes as well, so we are working with a sequence of Laplacians $\{\mathcal{W}_k\}_{k=1}^{\infty}$. That leads to a time-varying function $H_k(X) = \{H_k(X)\}_{k=1}^{\infty}$, where

$$
H_k(X) = \frac{1}{2} ||X - Y||^2 + \frac{R^2}{\epsilon_2} ||X \mathcal{W}_k||^2
$$

We are going to employ a decentralized minimization procedure. In order to do this, the gradient $\nabla H_k(X)$ should be computed in a decentralized setup.

$$
\nabla H_k(X) = X - Y + \frac{2R^2}{\epsilon_2} X \mathcal{W}
$$

Now recall the structure of $X, Y$ and $W$. Each of these quantities is a matrix of $R^{d \times n}$ with the $i$-th column stored at the $i$-th computational node. Consider $[\nabla H_k(X)]_i$ (i-th column of gradient).

$$
[\nabla H_k(X)]_i = [X]_i - [Y]_i + 2\frac{R^2}{\epsilon_2} [XW]_i
$$

Note that $[X]_i$ and $[Y]_i$ are held at node $i$, and $[XW]_i$ is computed as

$$
[XW]_i = \text{deg} \cdot [X]_i - \sum_{j \neq i, (i,j) \in E_k} [X]_j,
$$

where $E_k$ denotes the edge set of communication graph $\mathcal{G}_k$. Equation (19) means that $[XW]_i$ can be computed by agent $i$ via communication with its neighbours. Therefore, $[\nabla H_k(X)]_i$ can be computed locally on node $i$, which makes $\nabla H_k(X)$ available for decentralized computation.

We employ non-accelerated gradient descent on a time-varying function (18). The analysis of this procedure is performed in Section 3. First, we prove auxiliary lemmas.

Lemma 15.

(1) Let $X^*$ be the solution of (17). Then $X^* \in Y + K^\perp$. (2) Gradient descent applied to time-varying problem (18) with starting point $Y$ stays in $Y + K^\perp$.

Proof.

(1) Consider $\Delta X \in K$. Since $X^*$ is the solution of (17),

$$
\frac{R^2}{\epsilon_2} ||(X^* + \Delta X) \mathcal{W}||^2 + \frac{1}{2} ||(X^* + \Delta X) - Y||^2
\geq \frac{R^2}{\epsilon_2} ||X^* \mathcal{W}||^2 + \frac{1}{2} ||X^* - Y||^2
$$

$$
||X^* + \Delta X - Y||^2 \geq ||X^* - Y||^2
$$

Moreover, we can penalize the constraint $\Delta X \in K^\perp$.

(2) It is sufficient to show that for any $X \in Y + K^\perp$, the gradient of function in (18) lies in $K^\perp$. Consider some $\Delta X \in K^\perp$.

$$
\left\langle \nabla \frac{R^2}{\epsilon} ||X \mathcal{W}||^2 + \frac{1}{2} ||X - Y||^2 \right\rangle, \Delta X
$$

$$
= \left\langle \frac{2R^2}{\epsilon} - X \mathcal{W} + (X - Y), \Delta X \right\rangle
$$

$$
= \frac{2R^2}{\epsilon} \langle X, \Delta X \mathcal{W} \rangle + \langle X - Y, \Delta X \rangle = 0
$$

Lemma 16. Denote $X^* = \text{Proj}_K(Y)$ and let $X^*_W$ be the solution of (17). Then $||X^*_W - X^*||^2 \leq 4 \epsilon_2$. Proof. By Proposition 5, it holds

$$
||X_W - X^*||^2 \leq \frac{4}{\epsilon_2} ||\nabla \frac{1}{2} ||X - Y||^2 ||X = X^*||^2
$$

$$
= \frac{R^2}{\epsilon_2} \sqrt{\lambda^2_{max}(W)} \leq \frac{4}{\epsilon_2} ||X^* - Y|| \cdot \sqrt{\lambda^2_{max}(W)}
$$

Finally, using Lemmas 15 and 16 we establish the number of iterations for finding projection.

Theorem 17. After $N = O \left( \frac{\log \left( \frac{L^2 \lambda}{\epsilon \epsilon_2} \right)}{\epsilon \epsilon_2} \right)$ iterations (see (13) for definition of $\chi$), gradient descent on problem (18) with $\epsilon_2 = \frac{\lambda^2}{\epsilon \chi}$ yields $X_N$ such that

$$
||X_N - X^*||^2 \leq \epsilon_1
$$

Proof of Theorem 17 is provided in Appendix B.

4.5 Overall Complexity

Summarizing the results of Theorems 14 and 17, we get the final iteration complexity result.

Theorem 18. Algorithm 1 with $\epsilon_1 = \frac{\mu^2}{\gamma_0 \epsilon_1 \hat{L}_{max}}$ requires

$$
N = O \left( \frac{\mu^3 \chi \sqrt{\log \left( \frac{\lambda^2 \lambda_{max}(W)}{\epsilon \epsilon_2} \right)}}{\epsilon \epsilon_2} \right)
$$

communication steps, including sub-problem solution, to yield $X_N$ such that

$$
||X_N - X^*||^2 \leq \epsilon
$$
Remark 19. The convergence rate depends on $L_f$ and $\mu_f$ instead of $\mu_{\text{sum}} = \sum_{i=1}^n \mu_i$ and $L_{\text{sum}} = \sum_{i=1}^n L_i$. First, note that $\mu_f \geq \overline{\mu}$ and $L_f \leq \overline{L}$. Second, and most importantly, the ratio $L_{\text{sum}}/L_f$ may be of magnitude $n$, and the ratio $\mu_f/\mu_{\text{sum}}$ may be arbitrary large. We illustrate this observation with the following example.

$$f(x) = \frac{1}{2}(1 + \alpha)\|x\|^2, \alpha > 0;$$

$$f_i(x) = \frac{1}{2}x_i^2 + \frac{\alpha}{2n}\|x\|^2.$$ 

In this particular case, each $f_i(x)$ has $\mu_i = \alpha/n$ and $L_i = 1 + \alpha/n$, and therefore $L_{\text{sum}} = n + \alpha, \mu_{\text{sum}} = \alpha$. On the other hand, $\mu_f = L_f = 1 + \alpha$. Hence,

$$\frac{L_{\text{sum}}}{L_f} = \frac{n + \alpha}{1 + \alpha} \xrightarrow{\alpha \to +0} n,$$

$$\frac{\mu_f}{\mu_{\text{sum}}} = \frac{1 + \alpha}{\alpha} \xrightarrow{\alpha \to +0} \infty.$$ 

The bound obtained in Theorem 18 is based on $L_f/\mu_f$. The example above shows that using this ratio in the bound may be significantly better than using $L_{\text{sum}}/\mu_{\text{sum}}$.

5. NUMERICAL EXPERIMENTS

In this section, we provide numerical simulations of Algorithm 1 on logistic regression problem on LibSVM datasets (Chang and Lin (2011)). The objective function is defined as

$$f(x) = \frac{1}{m} \sum_{i=1}^m \log [1 + \exp(-(a_i, x) + b_i)],$$

where $a_i \in \mathbb{R}^d$ are training samples and $c_i \in \{0, 1\}$ are class labels. In decentralized scenario, the training dataset is distributed between the agents in the network.

One of the tuned parameters of Algorithm 1 is the number of inner iterations. On Figures 1 and 2, we illustrate different choices of this parameter, and Proj-GD-$k$ denotes projected gradient method with $k$ iterations on each subproblem. Moreover, we compare our algorithm to DIGing (Nedi et al. (2017)).

![Fig. 1. Random graph with 100 nodes, A9A dataset.](image)

![Fig. 2. Random graph with 100 nodes, W8A dataset.](image)

6. CONCLUSIONS AND FUTURE WORK

Our main result is based on a simple idea – running projected gradient method with inexact projections. This idea is applied to decentralized optimization on time-varying graphs. The proposed method incorporates two different algorithms: projected gradient descent and obtaining mean of values held by agents over the network. The whole procedure is shown to be robust to network changes since non-accelerated schemes are used both for outer and inner loops.

However, the question whether it is possible to employ an accelerated method either for finding projection or for running the outer loop remains open. Moreover, projection may be performed by a variety of algorithms, including randomized and asynchronous gossip algorithms (Boyd et al. (2006)). Investigation of new techniques for finding projection is left for future work.

REFERENCES

Arjevani, Y. and Shamir, O. (2015). Communication complexity of distributed convex learning and optimization. In Advances in neural information processing systems, 756–764.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). Parallel and distributed computation: numerical methods, volume 23. Prentice hall Englewood Cliffs, N.J.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. IEEE/ACM Trans. Netw., 14(SI), 2508–2530.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. Found. Trends Mach. Learn., 8(3-4), 231–357.

Chang, C.C. and Lin, C.J. (2011). Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3), 27.

Dvinskikh, D. and Gasnikov, A. (2019). Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. arXiv preprint arXiv:1904.09015.

Gasnikov, A. (2017). Universal gradient descent. arXiv preprint arXiv:1711.00394.

Gasnikov, A. (2018). Universal gradient descent. arXiv:1711.00394. [in Russian].

Hendrik, H., Bach, F., and Massoulié, L. (2018). Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. arXiv preprint arXiv:1810.02660.
Appendix A. PROOF OF THEOREM 14

Denote \( \Pi_k = \text{Proj}_{K}(X_k), \) \( X^* = \arg \min_{K} F(X), \) \( r_k = \|\Pi_k - \Pi^*\| = \|\Pi_k - X^*\|. \)
\[
r_{k+1}^2 = \|\Pi_k - \Pi^* - \gamma \text{Proj}_{K}(\nabla F(X_k))\|^2
- 2\gamma \langle \Pi_k - \Pi^*, \text{Proj}_{K}(\nabla F(X_k)) \rangle
+ \gamma^2 \|\text{Proj}_{K}(\nabla F(X_k))\|^2
\]
\[
= r_k^2 + \gamma^2 \|\text{Proj}_{K}(\nabla F(X_k))\|^2
- 2\gamma \langle \Pi_k - \Pi^*, \text{Proj}_{K}(\nabla F(X_k)) \rangle
\tag{A.1}
\]

(1) First, let us estimate \( \boxdot \). Note that for all \( \Pi \in K \), it holds
\[
\Pi = [\pi, \ldots, \pi]
\]
\[
\nabla F(\Pi) = [\nabla f_1(\pi), \ldots, \nabla f_n(\pi)]
\]

Moreover, for all \( X \in \mathbb{R}^{d \times n} \) it holds
\[
\text{Proj}_{K}(X) = \arg \min_{\mathcal{F}} \|Z - X\|^2 = [\pi, \ldots, \pi],
\]
where \( \pi = \frac{1}{n} \sum_{i=1}^{n} x_i \). In particular,
\[
\text{Proj}_{K}(\nabla F(\Pi)) = [\nabla f(\pi)/n, \ldots, \nabla f(\pi)/n]
\]

Now we can estimate \( \boxdot \) by Proposition 8. For brevity we introduce \( \tilde{\mu}_f = \mu_f/n, \tilde{L}_f = L_f/n. \)
\[
\begin{align*}
\langle \Pi_k - \Pi^*, \text{Proj}_{K}(\nabla F(\Pi_k)) - \text{Proj}_{K}(\nabla F(X^*)) \rangle \\
= n \cdot \langle \pi_k - \pi^*, \nabla f(\pi)/n - \nabla f(\pi^*)/n \rangle \\
= \langle \pi_k - \pi^*, \nabla f(\pi) - \nabla f(\pi^*) \rangle \\
\geq \frac{(n \tilde{\mu}_f)(n \tilde{L}_f)}{n \tilde{\mu}_f + n \tilde{L}_f} \|\pi_k - \pi^*\|^2 \\
+ \frac{1}{n \tilde{\mu}_f + n \tilde{L}_f} \|\nabla f(\pi_k) - \nabla f(\pi^*)\|^2 \\
= \gamma \frac{\tilde{\mu}_f \tilde{L}_f}{\tilde{\mu}_f + \tilde{L}_f} \|\Pi_k - \Pi^*\|^2 + \frac{1}{\tilde{\mu}_f + \tilde{L}_f} \|\text{Proj}_{K}(\nabla F(\Pi_k))\|^2
\end{align*}
\]

\[
(2) \text{ Let us employ (5) with } p = \frac{\tilde{\mu}_f + \tilde{L}_f}{\tilde{\mu}_f L_f} \text{ to estimate } \boxdot. \\
- 2\gamma \langle \Pi_k - \Pi^*, \text{Proj}_{K}(\nabla F(X_k)) - \text{Proj}_{K}(\nabla F(\Pi_k)) \rangle \\
\leq \gamma \frac{\tilde{\mu}_f \tilde{L}_f}{\tilde{\mu}_f + \tilde{L}_f} \|\Pi_k - \Pi^*\|^2 \\
+ \gamma \frac{\tilde{\mu}_f + \tilde{L}_f}{\tilde{\mu}_f L_f} \|\nabla F(X_k) - \nabla F(\Pi_k)\|^2 \\
\leq \gamma \frac{\tilde{\mu}_f \tilde{L}_f}{\tilde{\mu}_f + L_f} \|\Pi_k - \Pi^*\|^2 + \frac{\tilde{\mu}_f + \tilde{L}_f}{\tilde{\mu}_f L_f} L_{\max}^2 \epsilon_1
\]

Now we return to (A.1).
\[
r_{k+1}^2 \leq r_k^2 + \gamma^2 \|\text{Proj}_{K}(\nabla F(X_k))\|^2 \\
- 2\gamma \frac{\tilde{\mu}_f \tilde{L}_f}{\tilde{\mu}_f + L_f} r_k^2 + 2\gamma \frac{1}{\tilde{\mu}_f + L_f} \|\nabla \text{Proj}_{K}(\nabla F(\Pi_k))\|^2 \\
+ \gamma \frac{\tilde{\mu}_f \tilde{L}_f}{\tilde{\mu}_f + L_f} r_k^2 + \frac{\tilde{\mu}_f + \tilde{L}_f}{\tilde{\mu}_f L_f} L_{\max}^2 \epsilon_1 \tag{A.2}
\]

The sum of underlined terms may be estimated by setting \( \gamma \in \left(0, \frac{2}{\mu_f + L_f}\right)\) and using (6) with \( p = \gamma \frac{\tilde{\mu}_f + \tilde{L}_f}{\tilde{\mu}_f L_f} \in (0, 1). \)
\[ \gamma^2 \| \text{Proj}_K(\nabla F(X_k)) \|^2 - \frac{2\gamma}{\hat{\mu}_f + L_f} \| \nabla \text{Proj}_K(F(\Pi_k)) \|^2 \]
\[ = \frac{2\gamma}{\hat{\mu}_f + L_f} \left( \frac{\gamma(\hat{\mu}_f + \hat{L}_f)}{2} \| \text{Proj}_K(\nabla F(X_k)) \|^2 - \| \text{Proj}_K(\nabla F(\Pi_k)) \|^2 \right) \]
\[ \leq \frac{2\gamma}{\hat{\mu}_f + L_f} \cdot \frac{\gamma(\hat{\mu}_f + \hat{L}_f)}{2} \left( 1 - \frac{\gamma(\hat{\mu}_f + \hat{L}_f)}{2} \right)^{-1} \cdot \| \text{Proj}_K(\nabla F(X_k)) - \text{Proj}_K(\nabla F(\Pi_k)) \|^2 \]
\[ \leq \frac{2\gamma^2}{2 - \gamma(\hat{\mu}_f + \hat{L}_f)} \cdot L_{\max}^2 \varepsilon_1 \]

Finally, we return to (A.2), set \( \gamma = \frac{1}{\hat{\mu}_f + L_f} \) and estimate \( r_{k+1} \).
\[ r_{k+1}^2 \leq r_k^2 \left( 1 - \frac{\hat{\mu}_f \hat{L}_f}{\hat{\mu}_f + \hat{L}_f} \right) \]
\[ + \left( \frac{\hat{\mu}_f + \hat{L}_f}{\hat{\mu}_f L_f} + \frac{2\gamma^2}{2 - \gamma(\hat{\mu}_f + \hat{L}_f)} \right) \cdot L_{\max}^2 \varepsilon_1 \]
\[ = r_k^2 \left( 1 - \frac{\hat{\mu}_f \hat{L}_f}{(\hat{\mu}_f + \hat{L}_f)^2} + \frac{1}{\hat{\mu}_f L_f} + \frac{2}{(\hat{\mu}_f + \hat{L}_f)^2} \right) \cdot L_{\max}^2 \varepsilon_1 \]
\[ \leq r_k^2 \left( 1 - \frac{\hat{\mu}_f}{4L_f} \right) + \frac{3}{2\hat{\mu}_f L_f} \cdot L_{\max}^2 \varepsilon_1 \]

Analogously to Section 3, for \( r_k^2 \geq \frac{12}{\hat{L}_f^2} \cdot L_{\max}^2 \varepsilon_1 \) it holds
\[ r_{k+1}^2 \leq r_k^2 \left( 1 - \frac{\hat{\mu}_f \hat{L}_f}{8L_f} \right) . \]

After \( N \) iterations, we get
\[ \| \Pi_N - X^* \| = \| \Pi_N - \Pi^* \|^2 = r_N^2 \leq \frac{12}{\hat{\mu}_f^2} \cdot L_{\max}^2 \varepsilon_1 \]
Since \( \| X_N - \Pi_N \|^2 \leq \varepsilon_1 \),
\[ \| X_N - X^* \|^2 = \| X_N - \Pi_N \|^2 + \| \Pi_N - X^* \|^2 \]
\[ \leq \frac{12}{\hat{\mu}_f^2} \cdot L_{\max}^2 \varepsilon_1 + \varepsilon_1 \leq \frac{13}{\hat{\mu}_f^2} \cdot L_{\max}^2 \varepsilon_1 = \varepsilon , \]
which concludes the proof.

Appendix B. PROOF OF THEOREM 17

Gradient descent is run on a time-varying function \( H(X) = \{ H_k(X) \}_{k=1}^\infty \) with \( H_k \) is defined in (18). By Lemma 16, for each \( k \) it holds \( \| X^* - \arg \min \ H_k(X) \|^2 \leq 4\varepsilon_2 \). Moreover, each \( H_k(X) \) \( \mu_k \)-strongly convex on \( K^+ \) and \( L_k \)-smooth on \( \mathbb{R}^{d \times n} \), where
\[ \mu_k = 1 + \frac{2R^2}{\varepsilon_2} \lambda_{\min}^+(W_k) \geq 1 + \frac{2R^2}{\varepsilon_2} \theta_{\min} \]
\[ L_k = 1 + \frac{2R^2}{\varepsilon_2} \lambda_{\max}(W_k) \leq 1 + \frac{2R^2}{\varepsilon_2} \theta_{\max} \]
Therefore, time-varying function \( H(X) \) satisfies Assumption 7. Then, by Theorem 11, it follows that