Communication efficient privacy-preserving distributed optimization using adaptive differential quantization

Li, Qiongxiu; Heusdens, Richard; Christensen, Mads Græsbøll

Published in:
Signal Processing

DOI (link to publication from Publisher):
10.1016/j.sigpro.2022.108456

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Signal Processing 194 (2022) 108456

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Communication efficient privacy-preserving distributed optimization using adaptive differential quantization

Qiongxiu Li[a,*], Richard Heusdens[b], Mads Græsbøll Christensen[a]

[a] Audio Analysis Lab, CREATE, Aalborg University, Denmark
[b] Netherlands Defence Academy and Delft University of Technology, the Netherlands

ARTICLE INFO

Article history:
Received 27 August 2021
Revised 4 December 2021
Accepted 6 January 2022
Available online 8 January 2022

Keywords:
Distributed optimization
Quantization
Communication cost
Privacy
Information-theoretic
ADMM
PDMM

ABSTRACT

Privacy issues and communication cost are both major concerns in distributed optimization in networks. There is often a trade-off between them because the encryption methods used for privacy-preservation often require expensive communication overhead. To address these issues, we, in this paper, propose a quantization-based approach to achieve both communication efficient and privacy-preserving solutions in the context of distributed optimization. By deploying an adaptive differential quantization scheme, we allow each node in the network to achieve its optimum solution with a low communication cost while keeping its private data unrevealed. Additionally, the proposed approach is general and can be applied in various distributed optimization methods, such as the primal-dual method of multipliers (PDMM) and the alternating direction method of multipliers (ADMM). We consider two widely used adversary models, passive and eavesdropping, and investigate the properties of the proposed approach using different applications and demonstrate its superior performance compared to existing privacy-preserving approaches in terms of both accuracy and communication cost.

1. Introduction

With the emergence of interconnected or networked systems, distributed optimization is widely used to process its massive amount of data. As the primary computation units in these distributed networks are often personal devices, such as mobile phones and tablets [1,2], the underlying networked data often carry sensitive personal information, thus are private in nature. Furthermore, the available computational resources are also limited by the hardware and energy consumption, e.g., in wireless (acoustic) sensor networks [3–6]. As a consequence, novel distributed optimization tools are required that are able to address the privacy concern in a way that is efficient in terms of communication and computational resources.

Existing approaches mostly address the above challenges only partially. To achieve computationally lightweight solutions, noise insertion approaches, which add noise to obfuscate the private data, are widely used in the literature. These methods can be broadly classified into three classes. The first one is the class of differentially private distributed optimization approaches [7–13]. The main idea is to guarantee that the posterior guess of the private data is only slightly better than the prior guess. The downside of these algorithms is that the algorithm accuracy is degraded, i.e., they have an inherent trade-off between privacy and accuracy. The second class is that of secret-sharing based distributed optimization approaches [14,15] which deploy secret sharing to prevent privacy leakage, a technique used in secure multiparty computation [16,17]. Secret sharing works by splitting the private data into a number of so-called shares and distributes them over the nodes such that without a sufficient number of nodes cooperating the private data cannot be reconstructed. It, however, often suffers from high communication costs as the distribution and collection of shares requires extra communication rounds. The third class is the class of subspace perturbation based distributed optimization approaches [18–20] which, by inserting noise in a subspace determined by the graph topology, alleviates the privacy-accuracy trade-off without severely increasing the communication costs.

When considering the communication cost, aside from the number of times the communication channel is used, there is another critical parameter, namely the communication bandwidth or the corresponding bit-rate. The communication bandwidth is often omitted in privacy related approaches by assuming infinite precision. However, there is often a fundamental trade-off between privacy and communication cost in noise insertion type methods.

* Corresponding author.
E-mail addresses: qili@create.aau.dk (Q. Li), r.heusdens@mindef.nl, r.heusdens@tudelft.nl (R. Heusdens), mgc@create.aau.dk (M.G. Christensen).

https://doi.org/10.1016/j.sigpro.2022.108456

The reason for this is that a higher privacy level usually requires a larger amount of noise insertion, which, in turn, increases the entropy of the noise and thereby its bit-rate.

A typical way to save the communication bandwidth is to apply quantization schemes. However, existing quantization based distributed optimization schemes, e.g., [21–24], mostly consider the effects on the algorithm accuracy and convergence rate. The privacy primitive, however, is remained unexplored. In this paper, we aim to rectify this by linking these important parameters together, such that a communication efficient and privacy-preserving distribution optimization framework can be achieved.

1.1. Paper contributions

To the best of our knowledge, this is the first approach which provides information theoretical privacy guarantees for distributed optimization with quantization. The main idea is to exploit an adaptive differential quantization scheme in a particular way such that the communication cost is reduced while the private data is kept unrevealed. The proposed approach has a number of attractive properties:

- The accuracy of the optimization result is not compromised by considering both privacy and quantization.
- The overall communication cost is significantly reduced compared to existing privacy-preserving approaches including secret sharing, subspace perturbation, and differential privacy based approaches.
- It is generally applicable to various distributed optimizers such as ADMM, PDMM and similar algorithms.
- It provides privacy guarantees under two widely-considered adversary models, namely eavesdropping and passive adversary models.

1.2. Outlines and notation

The remaining part of the paper is organized as follows. Section 2 reviews fundamentals of distributed optimization. Section 3 introduces important concepts about privacy and then states the problem to be solved. Section 4 explains the reason why there is always a trade-off between privacy and communication bandwidth and Section 5 introduces the proposed approach to address it. Numerical results and comparisons with existing approaches are demonstrated in Section 6. Finally, the conclusion is given in Section 7.

We use the following notation throughout this paper. Lowercase letters x denote scalars, lowercase boldface letters $\boldsymbol{x}$ denote vectors, uppercase boldface letters $\boldsymbol{X}$ denote matrices. $\boldsymbol{x}_i$ and $\boldsymbol{X}_{ij}$ denote the ith and (i,j)th entry of the vector $\boldsymbol{x}$ and the matrix $\boldsymbol{X}$, respectively. Denote calligraphic letters $\mathcal{X}$ as sets and uppercase letters X denote random variables having realizations x. $H(X)$ and $h(X)$ denote the Shannon entropy and differential entropy of a random variable X, respectively.

2. Fundamentals

This section reviews necessary fundamentals for distributed optimization.

2.1. Distributed optimization over networks

We model a distributed network as a graph $\mathcal{G}=(\mathcal{N},\mathcal{E})$ where $\mathcal{N}=\{1,2,\ldots,n\}$ denotes the set of n nodes and $\mathcal{E}\subseteq\mathcal{N}\times\mathcal{N}$ denotes the set of m edges. Moreover, let $\mathcal{N}_i=\{j\,|\,(i,j)\in\mathcal{E}\}$ denote the set of neighboring nodes and $d_i=|\mathcal{N}_i|$ denote the degree (number of neighboring nodes) of node i. Many problems for example in statistical and machine learning fields can be formulated as a distributed convex optimization problem with constraints over the network [25]:

$$\begin{aligned}&\min_{\boldsymbol{x}}\sum_{i\in\mathcal{N}}f_i(\boldsymbol{x}_i),\\&\text{s.t. }\forall(i,j)\in\mathcal{E}:\boldsymbol{B}_{i|j}\boldsymbol{x}_i+\boldsymbol{B}_{j|i}\boldsymbol{x}_j=\boldsymbol{b}_{i,j},\end{aligned}\tag{1}$$

where $\boldsymbol{x}_i\in\mathbb{R}^u$ denotes the optimization variable of node i, $f_i:\mathbb{R}^u\mapsto\mathbb{R}\cup\{\infty\}$ denotes the local objective function which is assumed to be closed, convex and proper (CCP), and $\boldsymbol{B}_{i|j},\boldsymbol{B}_{j|i}$, being edge-related matrices (weights), and $\boldsymbol{b}_{i,j}\in\mathbb{R}^u$ denote the constraint imposed at edge $(i,j)\in\mathcal{E}$. For simplicity, we will assume $u=1$ (scalar variables), $\boldsymbol{B}_{i|j}=-\boldsymbol{B}_{j|i}=1$ if $i>j$ and $(i,j)\in\mathcal{E}$, and $\boldsymbol{b}_{i,j}=\boldsymbol{0}$. This corresponds to simple edge constraints of the form $\forall(i,j)\in\mathcal{E}:\boldsymbol{x}_i=\boldsymbol{x}_j$. The results, however, can straightforwardly be generalized to arbitrary dimensions and arbitrary (linear) edge constraints.

2.2. Distributed optimizers

To solve the problem in (1) in a decentralized manner where each node is only allowed to exchange information with its neighboring nodes, a number of distributed, iterative optimizers have been proposed, including ADMM [25] and PDMM [26–28]. It has been shown using monotone operator theory and operator splitting techniques that ADMM and PDMM are closely related [28] (see [29] for details on monotone operator theory). The update equations at node i at iteration $t=0,1,\ldots$ can be generally represented as

$$\boldsymbol{x}_i^{(t+1)}=\arg\min_{\boldsymbol{x}_i}\left(f_i(\boldsymbol{x}_i)+\textstyle\sum_{j\in\mathcal{N}_i}\boldsymbol{z}_{i|j}^{(t)}\boldsymbol{B}_{i|j}\boldsymbol{x}_i+\frac{cd_i}{2}\boldsymbol{x}_i^2\right),\tag{2}$$

$$\forall j\in\mathcal{N}_i:\boldsymbol{z}_{j|i}^{(t+1)}=\theta\boldsymbol{z}_{j|i}^{(t)}+(1-\theta)\left(\boldsymbol{z}_{i|j}^{(t)}+2c\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{(t+1)}\right),\tag{3}$$

where c is a constant for controlling the convergence rate. Each edge $e_k=(i,j)\in\mathcal{E}$, $k=1,2,\ldots,m$, is associated to two auxiliary variables $\boldsymbol{z}_k=\boldsymbol{z}_{i|j}$ and $\boldsymbol{z}_{k+m}=\boldsymbol{z}_{j|i}$, one for each node i and j, respectively. Stacking all auxiliary variables together we have $\boldsymbol{z}\in\mathbb{R}^{2m}$. $\theta\in[0,1)$ is a constant for controlling the averaging of the nonexpansive operators. For example, $\theta=0$ results in Peaceman-Rachford splitting (PDMM) while $\theta=1/2$ results in Douglas-Rachford splitting (ADMM).

3. Problem definition

In this section we first introduce important concepts regarding privacy-preservation and then define the problem to be solved and its evaluation metrics.

3.1. Privacy concern

In distributed optimization, sensitive personal information is often embedded in each node's local objective function $f_i(\boldsymbol{x}_i)$. The main reason is that the local objective function contains node-specific data as input and such data are often private in nature. As an example, consider a smart grid application. Assume each household/node in the network has its own power consumption data $\boldsymbol{s}_i$ and the goal is to compute the global average of the power consumption data, i.e., $n^{-1}\sum_{i\in\mathcal{N}}\boldsymbol{s}_i$. In this context the local objective function is given by $f_i(\boldsymbol{x}_i)=\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{s}_i\|_2^2$ and the overall problem setup can be formulated as follows:

$$\begin{aligned}\min_{\boldsymbol{x}}\quad&\sum_{i\in\mathcal{N}}\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{s}_i\|_2^2,\\\text{s.t.}\quad&\forall(i,j)\in\mathcal{E}:\boldsymbol{x}_i=\boldsymbol{x}_j.\end{aligned}\tag{4}$$

Note that the power consumption data $\boldsymbol{s}_i$ of each household, contained in the local objective function $f_i(\boldsymbol{x}_i)$, should be protected

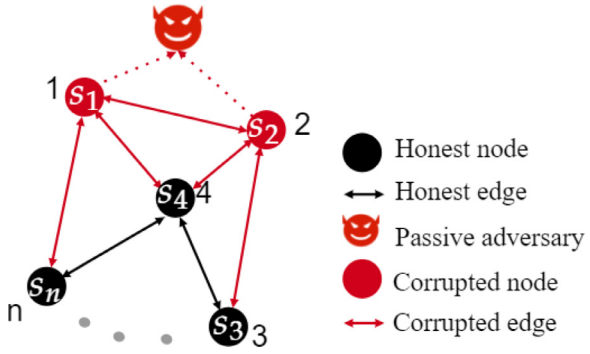


Fig. 1. Passive adversary.

from being revealed to others, as it can reveal information regarding the householders like their activities and even their health conditions (e.g., whether they are disabled) [30]. Hence, information regarding the local objective function $f_i(\boldsymbol{x}_i)$ is considered to be sensitive and should be protected from being revealed in the process of solving the optimization problem.

3.2. Adversary model

To analyze the privacy, we must specify an adversary model. The purpose of such a model is to quantify the system robustness in dealing with different security attacks. In this paper, we consider two types of adversary models: the passive and the eavesdropping model. The passive (also called semi-honest or honest-but-curious) adversary model is a classical model considered in distributed networks [31]. It works by a number of nodes colluding, referred to as corrupted nodes. The corrupted nodes are assumed to follow the algorithm instructions (called the protocol) but will share information together. We call an edge in the graph corrupted as long as there is one corrupted node at either end. All messages transmitted along such an edge will be known to the corrupted nodes, thus also to the passive adversary. See Fig. 1 for a toy example. Hence, the passive adversary will collect all information from the corrupted nodes to infer private data of the other non-colluding nodes (referred to as honest nodes).

The eavesdropping adversary is assumed to attack the system by listening to the messages transmitted along the communication channels, i.e., edges. This model receives little attention as it can be addressed by assuming all communication channels are securely encrypted such that the transmitted messages cannot be eavesdropped, e.g., secret sharing based approaches [14,15,32]. However, this assumption is particularly expensive to realize in distributed optimization applications, as a large number of iterations is often required before the algorithm converges.

We assume that the considered two adversaries can cooperate, i.e., they share information together with the aim of inferring the private data of the honest nodes.

3.3. Main requirements and related metrics

Putting things together, we now state the main requirements that communication efficient privacy-preserving distributed optimization should satisfy and introduce the related metrics.

1. **Output correctness**: Each node i should obtain the optimal solution to (1), denoted by $\boldsymbol{x}_i^*$, when the algorithm terminates. A typical way to quantify the output correctness is to adopt certain distance metrics to calculate the difference between the estimated output $\boldsymbol{x}^{(t)}$ and the optimum output $\boldsymbol{x}^*$. In this paper we use the overall mean square error (MSE) to quantify it, i.e., $\left\|\boldsymbol{x}^{(t)}-\boldsymbol{x}^*\right\|^2$.
2. **Communication cost**: After the algorithm execution, the cost of all communications should be as low as possible. The communication cost is given by $2mlT$, where T is the total number of iterations, $2m$ is the total amount of messages transmitted at each iteration (d_i per node and $\sum_{i\in\mathcal{N}} d_i = 2m$ in total), and l is the number of bits needed to represent each message.
3. **Individual privacy**: Each node's private information, embedded in $f_i(\boldsymbol{x}_i)$, should be protected under both eavesdropping and passive adversaries throughout the algorithm. As we are focusing on noise insertion approaches, we will focus on information-theoretical metrics to quantify the privacy. In the context of distributed processing, two commonly used metrics are the so-called ϵ-differential privacy and mutual information [33]. In this paper we choose mutual information as the individual privacy metric. The main reasons are as follows. (1) Mutual information has been proven effective in the context of privacy-preserving distributed processing [34], and has been applied in various applications [35–40]. (2) It is closely related to ϵ-differential privacy (see [41] for more details), and is easier to realize in practice [42–44]. (3) ϵ-differential privacy does not work if the private data is correlated [45].

4. Trade-off in noise insertion approaches

As mentioned in the introduction, existing computationally lightweight privacy-preserving methods mainly use the idea of noise insertion to achieve privacy. In this section, we aim to explain why there is typically a trade-off between privacy and communication bandwidth in such approaches. To do so, we will first explain a simple noise insertion scheme and then introduce how to compute the communication bandwidth, i.e., bit-rate, after applying quantization. Finally, we give an example to demonstrate this trade-off.

4.1. Additive noise insertion

Assume a scenario where a number of people, each having his/her own private data, would like to participate in a project which takes the private data of all these participants as input. Let s denote the private data held by you and you are reluctant to share your private data to others due to privacy concerns. The idea of noise insertion is to insert certain noise, denoted by r, to obfuscate the private data and then share the obfuscated data, denoted by s_r, to others. One of the most simple yet widely-used ways of noise insertion is to directly add the noise to the private data for protecting it from being revealed to others. This is referred to as additive noise insertion and can be expressed mathematically as

$$s_r = s + r. \tag{5}$$

Intuitively, a higher privacy level will be achieved if the obfuscated data s_r is less correlated with the private data s. We have the following result.

Proposition 1. *(Privacy guarantee for additive noise insertion) Let R and S be continuous random variables with variance $\sigma_R^2, \sigma_S^2 < \infty$, denoting the inserted noise and private data, respectively, and assume that R and S are statistically independent. Let $S_r = S + R$. Given an arbitrarily small $\delta > 0$, there exists $\beta > 0$ such that for $\sigma_R^2 \geq \beta$*

$$I(S; S_r) \leq \delta, \tag{6}$$

where $I(\cdot;\cdot)$ denotes mutual information. In the case that the noise R is Gaussian distributed, we have

$$\beta = \frac{\sigma_S^2}{2^{2\delta}-1}. \tag{7}$$

Proof. See [34, Proposition 1]. □

Hence, the more noise is inserted, the higher privacy level can be obtained. However, we remark that more noise will inevitably increase the noise entropy and thus requires a higher bit-rate (i.e., communication bandwidth). In what follows we will explain this in more detail.

4.2. Quantization and bit-rate

The main idea of quantization is to establish a mapping of the possibly continuous input data to a countable set of reproduction values, which is referred to as a codebook. More specifically, a quantizer divides the input domain into so-called quantization cells (i.e., Voronoi regions) where all elements within a cell are represented by a unique reproduction or code value. Let l denote the number of bits to represent the reproduction values, 2^l in total. Although there exists many different quantization schemes, we will introduce a simple yet effective one, namely uniform quantization. In this quantizer, all quantization cells have the same size, except for the cells at the boundary of the domain in the case of fixed bit-rate quantization. For example, a one-dimensional 2-bit uniform mid-rise quantizer with cell-width Δ will divide the input range into four regions, each represented by a unique code value. That is, the quantization cells are given by $(-\infty, -\Delta]$, $(-\Delta, 0]$, $(0, \Delta]$, (Δ, ∞) and represented by $-\frac{3\Delta}{2}$, $-\frac{\Delta}{2}$, $\frac{\Delta}{2}$, and $\frac{3\Delta}{2}$, respectively. Using a finite number of bits to quantize a continuous random variable will always introduce an error, or distortion. Intuitively, the fewer bits are used, the more distortion will occur. To estimate the number of bits required for transmitting a message, we determine the entropy since this gives a lower bound on the number of bits needed to represent the data. With a uniform quantizer, the entropy of a quantized continuous random variable X at sufficiently high rate can be approximated as [46]:

$$H(\hat{X}) \approx h(X) - \frac{1}{2}\log \Delta^2, \tag{8}$$

where $\hat{X}$ is the discrete random variable after quantizing X, $H(\hat{X})$ is the Shannon entropy of $\hat{X}$, and $h(X)$ is the differential entropy of X, assuming it exists. Since the differential entropy of a random variable with fixed variance σ^2 is upper bounded by $\frac{1}{2}\log\left(2\pi e\sigma^2\right)$, we have

$$H(\hat{X}) \leq \frac{1}{2}\log\left(\frac{2\pi e\sigma_X^2}{\Delta^2}\right). \tag{9}$$

4.3. Trade-off between privacy and bit-rate

Putting things together, we now proceed to analyze the amount of bits required for transmitting the obfuscated data S_r after considering quantization. By inspection of (7), we can see that given a desired privacy level δ and assuming that the inserted noise R is Gaussian distributed, $H(\hat{S}_r)$ can be upper bounded by

$$\begin{aligned} H(\hat{S}_r) &\leq \frac{1}{2}\log\left(\frac{2\pi e\sigma_{S_r}^2}{\Delta^2}\right) \\ &\overset{(a)}{=} \frac{1}{2}\log\left(\frac{2\pi e(\sigma_S^2+\sigma_R^2)}{\Delta^2}\right) \\ &\overset{(b)}{=} \frac{1}{2}\log\left(\frac{2\pi e(2^{2\delta}\sigma_S^2)}{(2^{2\delta}-1)\Delta^2}\right), \end{aligned} \tag{10}$$

where (a) holds as S and R are independent and (b) follows by setting σ_R^2 equal to β given by (7). By inspection of (10), we can see that the smaller the information leakage δ is, i.e., the higher the privacy level is, the higher the amount of bits for representing the quantized obfuscated data $\hat{S}_r$ will be. Clearly there is a trade-off between them. In Fig. 2 we give an example based on (10) to demonstrate this trade-off, where we set $\sigma_S^2 = 1$ and $\Delta = 10^{-5}$.

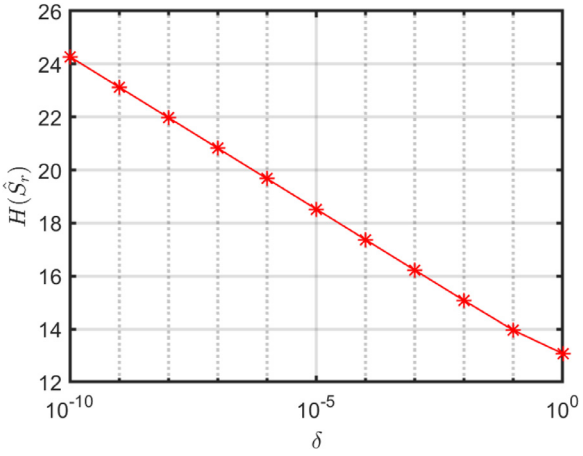


Fig. 2. Maximum number of bits for transmitting the obfuscated data S_r in terms of the privacy level δ when setting $\sigma_S^2 = 1$ and $\Delta = 10^{-5}$.

5. Proposed approach

After having explained why there is a trade-off between privacy and communication cost in noise insertion approaches, we now proceed to introduce the proposed approach for addressing it. The key idea is to adopt the adaptive differential quantization scheme of [47,48] to save communication cost without compromising both privacy and accuracy. More specifically, by quantizing (the difference of) the auxiliary variable $\boldsymbol{z}$, the communication cost can be significantly reduced. At the same time, the privacy of the proposed approach is guaranteed by making use of the initial (unknown) value of the auxiliary variable $\boldsymbol{z}^{(0)}$ which will serve as noise to protect the private data from being revealed to both passive and eavesdropping adversaries.

In what follows, we will introduce the proposed approach based on the concerned requirements mentioned in Section 3.3, i.e., individual privacy, output correctness and communication cost. We first introduce how to save communication cost using adaptive differential quantization and then explain what the effect of quantization is on the individual privacy using the above-mentioned noise insertion idea. Finally, we summarize the proposed approach and analyze the output correctness.

5.1. Communication efficiency through adaptive differential quantization

The main idea of applying adaptive differential quantization is based on the observation that for fixed point iterations the difference of successive iterations will converge to zero, which implies that the entropy of the difference of successive iterations will decrease to zero as the number of iteration increases. Motivated by this, the adaptive differential quantization scheme proposed in [47,48] quantizes the difference of the auxiliary variable with an adaptive cell-width decreasing with increasing iteration number. By doing so, low data-rate transmission between nodes can be achieved without compromising the accuracy of the algorithm.

With the adaptive differential quantization scheme, the process of the proposed approach is given as follows. At initialization $t = 0$, each node $i \in \mathcal{N}$ randomly initializes its auxiliary variables $\{\boldsymbol{z}_{i|j}^{(0)}\}_{j\in\mathcal{N}_i}$, sends the corresponding $\boldsymbol{z}_{i|j}^{(0)}$ to each and every neighboring node $j \in \mathcal{N}_i$, and updates $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{z}_{j|i}^{(1)}$ as

$$\boldsymbol{x}_i^{(1)} = \arg\min_{\boldsymbol{x}_i}\left(f_i(\boldsymbol{x}_i) + \textstyle\sum_{j\in\mathcal{N}_i}\boldsymbol{z}_{i|j}^{(0)}\boldsymbol{B}_{i|j}\boldsymbol{x}_i + \frac{cd_i}{2}\boldsymbol{x}_i^2\right), \tag{11}$$

$$\forall j \in \mathcal{N}_i : \boldsymbol{z}_{j|i}^{(1)} = \theta\boldsymbol{z}_{j|i}^{(0)} + (1-\theta)\left(\boldsymbol{z}_{i|j}^{(0)} + 2c\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{(1)}\right). \tag{12}$$

Let $\hat{\boldsymbol{z}}$ denote the quantized version of $\boldsymbol{z}$. At iteration $t \geq 1$, each node does not transmit the unquantized $\boldsymbol{z}_{j|i}^{(t)}$ to node j directly, instead it first defines the difference variable $\boldsymbol{v}^{(t)}$ as

$$\boldsymbol{v}^{(t)} \triangleq \begin{cases} \boldsymbol{z}^{(1)} - \boldsymbol{z}^{(0)}, & \text{if } t = 1, \\ \boldsymbol{z}^{(t)} - \hat{\boldsymbol{z}}^{(t-1)}, & \text{if } t > 1. \end{cases} \tag{13}$$

Let $Q(\cdot)$ denote the quantization operation. Applying quantization to the difference variable $\boldsymbol{v}^{(t)}$ we have

$$\hat{\boldsymbol{v}}^{(t)} = Q(\boldsymbol{v}^{(t)}) = \boldsymbol{v}^{(t)} + \boldsymbol{n}_{q,v^{(t)}}, \tag{14}$$

where $\boldsymbol{n}_{q,v^{(t)}}$ denotes the noise introduced by quantizing $\boldsymbol{v}^{(t)}$. After obtaining $\hat{\boldsymbol{v}}^{(t+1)}$, the quantized $\hat{\boldsymbol{z}}^{(t+1)}$ can be obtained by

$$\hat{\boldsymbol{z}}^{(t)} = \begin{cases} \boldsymbol{z}^{(0)} + \hat{\boldsymbol{v}}^{(1)}, & \text{if } t = 1, \\ \hat{\boldsymbol{z}}^{(t-1)} + \hat{\boldsymbol{v}}^{(t)}, & \text{if } t > 1. \end{cases} \tag{15}$$

Note that all $\{\hat{\boldsymbol{z}}^{(t)}\}_{t\geq 1}$ can be reconstructed when knowing $\boldsymbol{z}^{(0)}$ and the quantized $\{\hat{\boldsymbol{v}}^{(t)}\}_{t\geq 1}$ as

$$\forall t \geq 1 : \hat{\boldsymbol{z}}^{(t)} = \boldsymbol{z}^{(0)} + \sum_{\tau=1}^{t} \hat{\boldsymbol{v}}^{(\tau)}. \tag{16}$$

After constructing $\hat{\boldsymbol{z}}^{(t)}$, each node can update the local variables $\boldsymbol{x}_i^{(t+1)}$ and $\boldsymbol{z}_{j|i}^{(t+1)}$ using the quantized $\hat{\boldsymbol{z}}_{i|j}^{(t)}$ and $\hat{\boldsymbol{z}}_{j|i}^{(t)}$ from the previous iteration, i.e.,

$$\boldsymbol{x}_i^{(t+1)} = \arg\min_{\boldsymbol{x}_i} \left(f_i(\boldsymbol{x}_i) + \textstyle\sum_{j\in\mathcal{N}_i} \hat{\boldsymbol{z}}_{i|j}^{(t)} \boldsymbol{B}_{i|j} \boldsymbol{x}_i + \frac{cd_i}{2} \boldsymbol{x}_i^2 \right), \tag{17}$$

$$\forall j \in \mathcal{N}_i : \boldsymbol{z}_{j|i}^{(t+1)} = \theta \hat{\boldsymbol{z}}_{j|i}^{(t)} + (1-\theta)\left(\hat{\boldsymbol{z}}_{i|j}^{(t)} + 2c\boldsymbol{B}_{i|j} \boldsymbol{x}_i^{(t+1)} \right). \tag{18}$$

Overall, we conclude that all messages that need to be transmitted are the initialized $\boldsymbol{z}^{(0)}$ and the quantized $\{\hat{\boldsymbol{v}}^{(t)}\}_{t\geq 1}$. Because $\{\hat{\boldsymbol{z}}^{(t)}\}_{t\geq 1}$ can be computed using $\boldsymbol{z}^{(0)}$ and $\{\hat{\boldsymbol{v}}^{(t)}\}_{t\geq 1}$, $\{\boldsymbol{x}^{(t)}\}_{t\geq 1}$ can be computed using $\boldsymbol{z}^{(0)}$ and $\{\hat{\boldsymbol{z}}^{(t)}\}_{t\geq 1}$ using (11) and (17).

5.2. Privacy preservation based on additive noise insertion

Having introduced how to reduce the communication bandwidth, we now proceed to explain how to guarantee privacy in the context of adaptive differential privacy. Motivated by the idea of using additive noise insertion to achieve privacy-preservation, instead of inserting extra noise we propose to make use of the auxiliary variable $\boldsymbol{z}$ as noise. Indeed, with the help of adaptive differential quantization, we only need the initialized $\boldsymbol{z}_{i|j}^{(0)}$ to serve as noise. More specifically, each nodes only needs to initialize its own auxiliary variables $\{\boldsymbol{z}_{i|j}^{(0)}\}_{j\in\mathcal{N}_i}$ with distributions having large variances depending on the desired privacy level (see Proposition 1). The details of privacy analysis are given as follows.

Let $\partial f_i(x)$ denote the subdifferential of f_i at x. By inspection of (17), for $t \geq 1$ the updates $\boldsymbol{x}_i^{(t)}$ satisfy

$$0 \in \partial f_i(\boldsymbol{x}_i^{(t+1)}) + \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j} \hat{\boldsymbol{z}}_{i|j}^{(t)} + cd_i \boldsymbol{x}_i^{(t+1)}. \tag{19}$$

We can see that the private data is only contained in $\partial f_i(\boldsymbol{x}_i^{(t+1)})$. As a consequence, the goal of the privacy analysis is to see what information regarding $\partial f_i(\boldsymbol{x}_i^{(t+1)})$ is revealed during the iterations. Note that for $t = 0$ we have $0 \in \partial f_i(\boldsymbol{x}_i^{(1)}) + \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j} \boldsymbol{z}_{i|j}^{(0)} + cd_i \boldsymbol{x}_i^{(1)}$.

For simplicity, assume $\boldsymbol{B}_{i|j} = 1$ for all $j \in \mathcal{N}_i$. Denote $\mathcal{N}_c$ and $\mathcal{N}_h$ as the set of corrupted nodes and honest nodes, respectively. Let $\mathcal{N}_{i,c} = \mathcal{N}_i \cap \mathcal{N}_c$ and $\mathcal{N}_{i,h} = \mathcal{N}_i \cap \mathcal{N}_h$ denote the set of the corrupted and honest neighbors of the node i, respectively. In addition, we assume a worse case scenario where each honest node has at least one corrupted neighboring node, i.e., $\mathcal{N}_{i,c} \neq \emptyset$. Combining (13) and (14) we conclude that $\hat{\boldsymbol{v}}^{(t)} - \boldsymbol{n}_{q,v^{(t)}} = \boldsymbol{z}^{(t)} - \hat{\boldsymbol{z}}^{(t-1)}$, so that (15) can be expressed as

$$\hat{\boldsymbol{z}}^{(t)} = \boldsymbol{z}^{(t)} + \boldsymbol{n}_{q,v^{(t)}}. \tag{20}$$

For node $k \in \mathcal{N}_{i,c}$, using (18) and (20), we can express the left-hand side of (19) as

$$\partial f_i(\boldsymbol{x}_i^{(t+1)}) + \sum_{j\in\mathcal{N}_i} \hat{\boldsymbol{z}}_{i|j}^{(t)} + \frac{d_i}{2(1-\theta)} \left(\hat{\boldsymbol{z}}_{k|i}^{(t+1)} - \boldsymbol{n}_{q,v_{k|i}^{(t+1)}} - \theta \hat{\boldsymbol{z}}_{k|i}^{(t)} - (1-\theta)\hat{\boldsymbol{z}}_{i|k}^{(t)} \right). \tag{21}$$

To quantify the amount of information about the private data $\partial f_i(\boldsymbol{x}_i^{(t+1)})$ learned by the adversaries, we must first inspect what information is available to them. We first consider the passive adversary. As the passive adversary can collect all the information available to the corrupted nodes, it has the following knowledge:

$$\left\{ \boldsymbol{x}_i^{(t)} \right\}_{i\in\mathcal{N}_c, t\geq 1} \cup \left\{ \boldsymbol{z}_{i|j}^{(0)}, \hat{\boldsymbol{v}}_{i|j}^{(t+1)} \right\}_{(i,j)\in\mathcal{E}_c, t\geq 0},$$

where $\mathcal{E}_c = \{(i,j) \in \mathcal{E}, (i,j) \notin \mathcal{N}_h \times \mathcal{N}_h\}$ denotes the set of corrupted edges. With the above knowledge, the passive adversary is able to compute both $\sum_{j\in\mathcal{N}_{i,c}} \hat{\boldsymbol{z}}_{i|j}^{(t)}$ using (16) and $\frac{d_i}{2(1-\theta)} \left(\hat{\boldsymbol{z}}_{k|i}^{(t+1)} - \theta \hat{\boldsymbol{z}}_{k|i}^{(t)} - (1-\theta)\hat{\boldsymbol{z}}_{i|k}^{(t)} \right)$ in (21). After deducing these known terms, (21) reduces to

$$\partial f_i(\boldsymbol{x}_i^{(t+1)}) + \sum_{j\in\mathcal{N}_{i,h}} \hat{\boldsymbol{z}}_{i|j}^{(t)} - \left\{ \frac{d_i}{2(1-\theta)} \boldsymbol{n}_{q,v_{k|i}^{(t+1)}} \right\}_{k\in\mathcal{N}_{i,c}}. \tag{22}$$

Next, we consider the eavesdropping adversary. As mentioned in Section 3.2, the expense for securely encrypting the communication channels is very high. In order to minimize this expense, we propose to use secure channel encryption only once. More specifically, no channel encryption is involved except for transmitting $\boldsymbol{z}^{(0)}$ during the initialization step. As a consequence, the eavesdropping adversary can listen to all transmitted messages after initialization, i.e.,

$$\left\{ \hat{\boldsymbol{v}}_{i|j}^{(t)} \right\}_{(i,j)\in\mathcal{E}, t\geq 1},$$

note that it does not have knowledge about $\boldsymbol{z}_{i|j}^{(0)}$. Based on (16), we can, therefore, deduce $\sum_{\tau=1}^{t} \hat{\boldsymbol{v}}_{i|j}^{(\tau)}$ from $\hat{\boldsymbol{z}}_{i|j}^{(t)}$ in (22) as it is known to the eavesdropping adversary. Consequently, we conclude that all what the passive and eavesdropping adversaries observe about the honest node i is given by

$$\partial f_i(\boldsymbol{x}_i^{(t+1)}) + \sum_{j\in\mathcal{N}_{i,h}} \boldsymbol{z}_{i|j}^{(0)} - \left\{ \frac{d_i}{2(1-\theta)} \boldsymbol{n}_{q,v_{k|i}^{(t+1)}} \right\}_{k\in\mathcal{N}_{i,c}} \tag{23}$$

where the last term $\{\boldsymbol{n}_{q,v_{k|i}^{(t+1)}}\}_{j\in\mathcal{N}_{i,c}}$ will converge to the all-zero vector as the iterations proceed. If node i has at least one honest neighbor, i.e., $\mathcal{N}_{i,h} \neq \emptyset$, the term $\sum_{j\in\mathcal{N}_{i,h}} \boldsymbol{z}_{i|j}^{(0)}$ can be considered as noise. Hence, we can, by Proposition 1, protect the private data $\partial f_i(\boldsymbol{x}_i^{(t+1)})$ from being revealed by making the variance of $\boldsymbol{z}^{(0)}$ sufficiently large at the initialization step. Therefore, arbitrarily small information leakage regarding $\partial f_i(\boldsymbol{x}_i^{(t+1)})$ can be achieved at every iteration.

5.2.1. Privacy discussion

Some remarks are in place here. We have concluded that arbitrary privacy levels of the proposed approach can be achieved by controlling the variance of $\boldsymbol{z}^{(0)}$. As we are focusing in optimization algorithms which will converge, one immediate question to ask is *"as the iterations proceed, will the variance of the auxiliary variable decrease, thereby (partly) revealing the private information?"* In the

following we will show that this will not happen because the variance of the auxiliary variables $\boldsymbol{z}_{i|j}^{(t)}$ is lower bounded, the bound being dependent on $\boldsymbol{z}^{(0)}$.

We first express (2) and (3) compactly as

$$\boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}} \left(f(\boldsymbol{x}) + (\boldsymbol{z}^{(t)})^{\top}(\boldsymbol{C}\boldsymbol{x}) + \frac{c}{2}\|\boldsymbol{C}\boldsymbol{x}\|_2^2 \right), \tag{24}$$

$$\boldsymbol{z}^{(t+1)} = \theta \boldsymbol{z}^{(t)} + (1-\theta)(\boldsymbol{P}\boldsymbol{z}^{(t)} + 2c\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t+1)}), \tag{25}$$

where $\boldsymbol{C} = [\boldsymbol{B}_+^{\top}, \boldsymbol{B}_-^{\top}]^{\top} \in \mathbb{R}^{2m\times n}$, and $\boldsymbol{B}_+$ and $\boldsymbol{B}_-$ are the matrices containing only the positive and negative entries of $\boldsymbol{B}$, respectively. $\boldsymbol{P} \in \mathbb{R}^{2m\times 2m}$ denotes a permutation matrix which exchanges the first m rows and last m rows of the matrix it operates on. Consider two successive z-updates:

$$\boldsymbol{z}^{(t+2)} = \theta \boldsymbol{z}^{(t+1)} + (1-\theta)\left(\boldsymbol{P}\boldsymbol{z}^{(t+1)} + 2c\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t+2)}\right) \tag{26}$$

$$\begin{aligned} &= \left(\theta^2 + (1-\theta)^2\right)\boldsymbol{z}^{(t)} + 2\theta(1-\theta)\boldsymbol{P}\boldsymbol{z}^{(t)} \\ &\quad + 2c(1-\theta)\left((1-\theta)\boldsymbol{C}\boldsymbol{x}^{(t+1)} + \theta\boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t+1)} + \boldsymbol{P}\boldsymbol{C}\boldsymbol{x}^{(t+2)}\right), \end{aligned} \tag{27}$$

where we used that fact that $\boldsymbol{P}^2 = \boldsymbol{I}_{2m}$ with $\boldsymbol{I}_{2m}$ the identity matrix in $\mathbb{R}^{2m}$. Let $\Psi = \text{ran}(\boldsymbol{C}) + \text{ran}(\boldsymbol{P}\boldsymbol{C})$ so that $\Psi^{\perp} = \ker(\boldsymbol{C}^T) \cap \ker((\boldsymbol{P}\boldsymbol{C})^T)$. Since $[\boldsymbol{C}\ \boldsymbol{P}\boldsymbol{C}] \in \mathbb{R}^{2m\times 2n}$ can be interpreted as an incidence matrix of a new bipartite graph with $2n$ nodes and $2m$ edges [20], we conclude that $\dim(\Psi) \leq 2n-1$, assuming $m \geq n$. Therefor, $\Psi^{\perp}$ is non-empty as long as the number of edges is larger than or equal to the number of nodes in the network. To get more insight in the z-updates, we need the following lemma.

Lemma 5.1. *Let $\boldsymbol{x} \in \Psi$ and $\boldsymbol{y} \in \Psi^{\perp}$. Then $\boldsymbol{P}\boldsymbol{x} \in \Psi$ and $\boldsymbol{P}\boldsymbol{y} \in \Psi^{\perp}$.*

Proof. Since $\Psi = \text{ran}(\boldsymbol{C}) + \text{ran}(\boldsymbol{P}\boldsymbol{C})$, $\boldsymbol{P}\boldsymbol{x} \in \Psi$. Moreover, $\forall \boldsymbol{x} \in \Psi$: $(\boldsymbol{P}\boldsymbol{y}, \boldsymbol{x}) = (\boldsymbol{y}, \boldsymbol{P}^*\boldsymbol{x}) = (\boldsymbol{y}, \boldsymbol{P}\boldsymbol{x}) = 0$ and thus $\boldsymbol{P}\boldsymbol{y} \in \Psi^{\perp}$. □

Let $\boldsymbol{z}^{(0)} \in \Psi$ and thus $\boldsymbol{P}\boldsymbol{z}^{(0)} \in \Psi$ by Lemma 5.1. By inspection of (26) and (27), we conclude that $\boldsymbol{z}^{(t)} \in \Psi$ for all t. However, if we initialize $\boldsymbol{z}^{(0)}$ randomly, without restricting it to belong to Ψ, it has been shown in [28] that only the component $\Pi_{\Psi}\boldsymbol{z}^{(t)}$ will converge to a fixed point as $t \to \infty$, regardless of the initialization $\boldsymbol{z}^{(0)}$. As for $(\boldsymbol{I}_{2m} - \boldsymbol{\Pi}_{\Psi})\boldsymbol{z}^{(t)} = \boldsymbol{z}_{\Psi^{\perp}}^{(t)}$, the orthogonal projection of $\boldsymbol{z}^{(t)}$ onto $\Psi^{\perp}$, we have the following results.

Theorem 5.1. *Given the iterates (24) and (25). Then*

$$\boldsymbol{z}_{\Psi^{\perp}}^{(t)} = \frac{1}{2}\left(\boldsymbol{z}_{\Psi^{\perp}}^{(0)} + \boldsymbol{P}\boldsymbol{z}_{\Psi^{\perp}}^{(0)}\right) + \frac{1}{2}(2\theta-1)^t\left(\boldsymbol{z}_{\Psi^{\perp}}^{(0)} - \boldsymbol{P}\boldsymbol{z}_{\Psi^{\perp}}^{(0)}\right). \tag{28}$$

Proof. See Appendix A. □

Lemma 5.2. *Let $\boldsymbol{\Pi}_{\Psi}$ and $\boldsymbol{\Pi}_{\Psi^{\perp}} = \boldsymbol{I}_{2m} - \boldsymbol{\Pi}_{\Psi}$ denote the projection onto Ψ and $\Psi^{\perp}$, respectively. Then*

$$\boldsymbol{\Pi}_{\Psi}\boldsymbol{P} = \boldsymbol{P}\,\boldsymbol{\Pi}_{\Psi},$$

$$\boldsymbol{\Pi}_{\Psi^{\perp}}\boldsymbol{P} = \boldsymbol{P}\,\boldsymbol{\Pi}_{\Psi^{\perp}}.$$

Proof. Let $\boldsymbol{z} = \boldsymbol{z}_{\Psi} + \boldsymbol{z}_{\Psi^{\perp}}$, where $\boldsymbol{z}_{\Psi} = \boldsymbol{\Pi}_{\Psi}\boldsymbol{z}$ and $\boldsymbol{z}_{\Psi^{\perp}} = \boldsymbol{\Pi}_{\Psi^{\perp}}\boldsymbol{z}$, the projection of $\boldsymbol{z}$ onto Ψ and $\Psi^{\perp}$, respectively. Then $\boldsymbol{P}\boldsymbol{z}_{\Psi} \in \Psi$ and $\boldsymbol{P}\boldsymbol{z}_{\Psi^{\perp}} \in \Psi^{\perp}$ by Lemma 5.1, so that $\boldsymbol{\Pi}_{\Psi}\boldsymbol{P}\boldsymbol{z} = \boldsymbol{\Pi}_{\Psi}\boldsymbol{P}(\boldsymbol{z}_{\Psi} + \boldsymbol{z}_{\Psi^{\perp}}) = \boldsymbol{P}\boldsymbol{z}_{\Psi} = \boldsymbol{P}\,\boldsymbol{\Pi}_{\Psi}\boldsymbol{z}$. The statement that $\boldsymbol{\Pi}_{\Psi^{\perp}}\boldsymbol{P} = \boldsymbol{P}\,\boldsymbol{\Pi}_{\Psi^{\perp}}$ follows trivially. □

Corollary 5.1. *Given the iterates (24) and (25). Let $\mathrm{E}\left(\boldsymbol{Z}^{(0)}\boldsymbol{Z}^{(0)^T}\right) = \sigma^2\boldsymbol{I}_{2m}$ denote the covariance matrix of $\boldsymbol{Z}^{(0)}$. Then*

$$\mathrm{E}\left(\boldsymbol{Z}_{\Psi^{\perp}}^{(t)}\boldsymbol{Z}_{\Psi^{\perp}}^{(t)^T}\right) = \boldsymbol{\Pi}_{\Psi^{\perp}}\left(\frac{\sigma^2}{2}\left((\boldsymbol{I}_{2m} + \boldsymbol{P}) + |2\theta-1|^{2t}(\boldsymbol{I}_{2m} - \boldsymbol{P})\right)\right). \tag{29}$$

Proof. Since $\boldsymbol{Z}^{(0)}$ is drawn at random, the result follows from (28), Lemma 5.2 and the fact that $\boldsymbol{\Pi}_{\Psi^{\perp}}\boldsymbol{\Pi}_{\Psi^{\perp}}^T = \boldsymbol{\Pi}_{\Psi^{\perp}}$. □

By inspection of (29) we conclude that

$$\mathrm{E}\left(\boldsymbol{Z}_{\Psi^{\perp}}^{(t)}\boldsymbol{Z}_{\Psi^{\perp}}^{(t)^T}\right) = \begin{cases} \boldsymbol{\Pi}_{\Psi^{\perp}}\sigma^2, & \text{if } \theta = 0 \\ \frac{1}{2}(\boldsymbol{\Pi}_{\Psi^{\perp}} + \boldsymbol{P}\boldsymbol{\Pi}_{\Psi^{\perp}})\sigma^2, & \text{if } \theta = 0.5 \end{cases} \tag{30}$$

$$\mathrm{E}\left(\boldsymbol{Z}_{\Psi^{\perp}}^{(t)}\boldsymbol{Z}_{\Psi^{\perp}}^{(t)^T}\right) \to \frac{1}{2}(\boldsymbol{\Pi}_{\Psi^{\perp}} + \boldsymbol{P}\boldsymbol{\Pi}_{\Psi^{\perp}})\sigma^2, \text{ if } \theta \in (0,1), \theta \neq 0.5.$$

Hence, the lower bound on $\mathrm{E}\left(\boldsymbol{Z}_{\Psi^{\perp}}^{(t)}\boldsymbol{Z}_{\Psi^{\perp}}^{(t)^T}\right)$ is dependent on σ^2. By increasing σ^2 we can achieve a higher variance, and thus the privacy level.

In addition, the fact that $\boldsymbol{z}^{(t)}$ depends on $\boldsymbol{z}_{\Psi^{\perp}}^{(0)}$ will not prevent the optimization variable $\boldsymbol{x}^{(t)}$ to converge to $\boldsymbol{x}^*$. Indeed, since $(\boldsymbol{\Pi}_{\Psi^{\perp}}\boldsymbol{z}^{(t)})^{\top}\boldsymbol{C}\boldsymbol{x}^{(t)} = \boldsymbol{0}$, we conclude, by inspection of (24), that the output correctness will not be affected by $\boldsymbol{z}_{\Psi^{\perp}}^{(0)}$.

Similar results also hold for the case of quantization. By inspecting (16), we see that besides $\boldsymbol{z}^{(0)}$, the quantized variable $\hat{\boldsymbol{v}}^{(t)}$ will have a non-zero component in $\Psi^{\perp}$, i.e., $\Pi_{\Psi^{\perp}}\hat{\boldsymbol{v}}^{(t)} \neq 0$. The variance of this component, however, will eventually vanish as the iterations proceed.

Summarizing, we conclude that by simply letting each node i randomly initialize its $\{\boldsymbol{z}_{i|j}^{(0)}\}_{j\in\mathcal{N}_i}$ with a distribution having sufficiently large variance, the privacy of honest node $i \in \mathcal{N}_h$ is protected against both passive and eavesdropping adversaries as long as:

1. There is at least one honest neighbor. That is, $\mathcal{N}_{i,h} \neq \emptyset$.
2. The communication channels are securely encrypted in the initialization phase when transmitting $\boldsymbol{z}^{(0)}$.

Algorithm 1 shows the details of the proposed algorithm.

5.3. Output correctness analysis

We now analyze the output correctness of the proposed approach. In [48,49] it has been shown that if the sequence $\{\boldsymbol{n}_{q,v^{(t)}}\}_{t\geq 0}$ is finitely summable, then Douglas-Rachford splitting will convergence to a fixed point $\boldsymbol{x}^*$ which is the solution to (1). In addition, in [48] it was shown that if the worst case rate of decrease of $\|\boldsymbol{v}^{(t)}\|^2$ is known, the same decrease in cell width can be implemented to maintain a fixed bit rate for the quantization, whilst simultaneously ensuring that the sequence $(\boldsymbol{n}_{q,v^{(t)}})_{t\in\mathbb{N}}$ is finitely summable. Although the same conclusion is not necessarily true for PDMM or Peaceman-Rachford splitting, in practice it is observed that the same result holds. As an example, for a geometrically converging sequence with factor γ, we can choose the cell-width as $\Delta^{(t)} = \gamma^t\Delta^{(0)}$.

Ignoring overflow errors, we then have that $|\boldsymbol{n}_{q,v^{(t)}}| \leq \Delta^{(t)}/2 = \gamma^t\Delta^{(0)}/2$, so that

$$\sum_{t=0}^{\infty}|\boldsymbol{n}_{q,v^{(t)}}| \leq \frac{\Delta^{(0)}}{2}\sum_{t=0}^{\infty}\gamma^t = \frac{\Delta^{(0)}}{2}\frac{1}{1-\gamma} < \infty,$$

for $\gamma < 1$ and we conclude that the output correctness requirement is satisfied. Note that the geometric convergence factor γ can be computed in general when we have information about strong monotonicity and the Lipschitz constant of the objective function [51].

6. Numerical results

In this section, we will present simulation results for the proposed approach and compare this with existing approaches. We simulated a geometric network with $n = 30$ nodes where every two nodes are allowed to transmit messages if their distance is within a radius of $\sqrt{\frac{2\log(n)}{n}}$, as this condition ensures that the corresponding graph is connected with high probability [52].

Input : Initialized auxiliary variables: $\{\boldsymbol{z}_{i|j}^{(0)}\}_{(i,j)\in\mathcal{E}}$
Initial quantization cell-width: $\Delta^{(0)}$
Number of bits of quantizer: l
Rate of growth: γ

Output : Optimum optimization solutions: $\{\boldsymbol{x}_i^*\}_{i\in\mathcal{N}}$

Initialization: Based on the desired privacy level, each node $i \in \mathcal{N}$, initializes $\{\boldsymbol{z}_{i|j}^{(0)}\}_{j\in\mathcal{N}_i}$ with a distribution having large variance (e.g., (7) in Proposition 1).

if $\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_i^*\|^2 > \mathit{threshold}$ **then**

  if $t = 0$ **then**

    Update $\boldsymbol{x}_i^{(1)}$ using (11).

    Receive $\boldsymbol{z}_{j|i}^{(0)}$ from neighbors $j \in \mathcal{N}_i$ through securely encrypted channels [50].

    Update $\{\boldsymbol{z}_{j|i}^{(1)}\}_{j\in\mathcal{N}_i}$ using (12).

  end

  if $t \geq 1$ **then**

    Receive $\hat{\boldsymbol{v}}_{i|j}^{(t)}$ from neighbors $j \in \mathcal{N}_i$ through non-securely encrypted channels.

    Update $\{\hat{\boldsymbol{z}}_{i|j}^{(t)}\}_{j\in\mathcal{N}_i}$ using (15).

    Update $\boldsymbol{x}_i^{(t+1)}$ using (17), $\{\boldsymbol{z}_{j|i}^{(t+1)}\}_{j\in\mathcal{N}_i}$ using (18).

  end

  Compute $\{\boldsymbol{v}_{j|i}^{(t+1)}\}_{j\in\mathcal{N}_i}$ using (13).

  Quantize $\{\boldsymbol{v}_{j|i}^{(t+1)}\}_{j\in\mathcal{N}_i}$ with a l-bit uniform quantizer having cell-width $\Delta^{(t+1)} = \gamma^t\Delta^{(0)}$.

  Transmit $\hat{\boldsymbol{v}}_{j|i}^{(t+1)}$ to each neighbor $j \in \mathcal{N}_i$.

end

Algorithm 1. Communication efficient privacy-preserving distributed optimization using adaptive differential quantization. This reference is cited in algoirhtm [50].

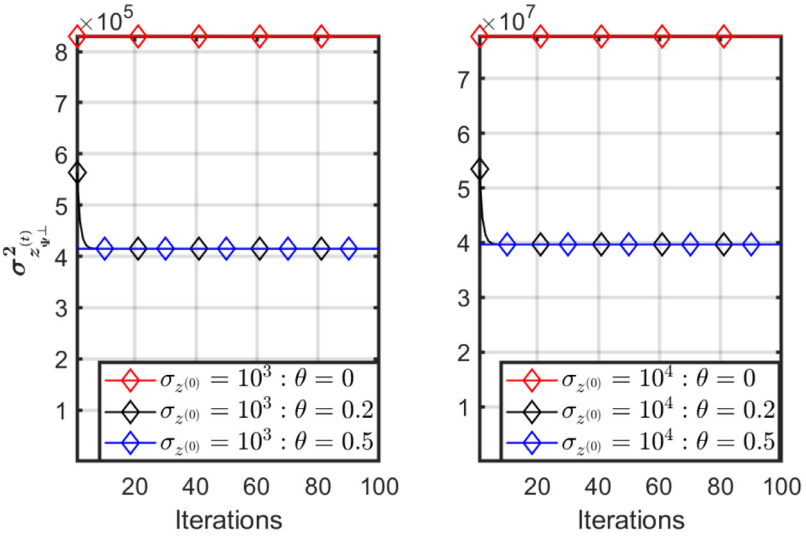
6.1. Performance of the proposed approach

We use two applications to test the performance of the proposed approach and exemplify its use: distributed average consensus and distributed least squares, as they have been intensively investigated in the literature [15,18,19,32,53–59]. The detailed problem formulation of distributed average consensus is already introduced in (4). As for distributed least squares, assume each node has partial knowledge of a linear system (assuming overdetermined) including an input observation, denoted as $\boldsymbol{Q}_i \in \mathbb{R}^{p_i\times u}$, $p_i > u$, and a decision vector, denoted as $\boldsymbol{y}_i \in \mathbb{R}^{p_i}$. Stacking the partial knowledge together we denote $\boldsymbol{Q} = [\boldsymbol{Q}_1^\top, \ldots, \boldsymbol{Q}_n^\top]^\top \in \mathbb{R}^{P_n\times u}$ and $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_n^\top]^\top \in \mathbb{R}^{P_n}$, where $P_n = \sum_{i\in\mathcal{N}} p_i$. The goal of privacy-preserving distributed least squares is to allow each node to achieve the global optimum solution $\forall i \in \mathcal{N}$, $\boldsymbol{x}_i^* = (\boldsymbol{Q}^\top\boldsymbol{Q})^{-1}\boldsymbol{Q}^\top\boldsymbol{y} \in \mathbb{R}^u$, without revealing its private data, i.e., $\boldsymbol{Q}_i, \boldsymbol{y}_i$. With distributed optimization this problem can be formulated as

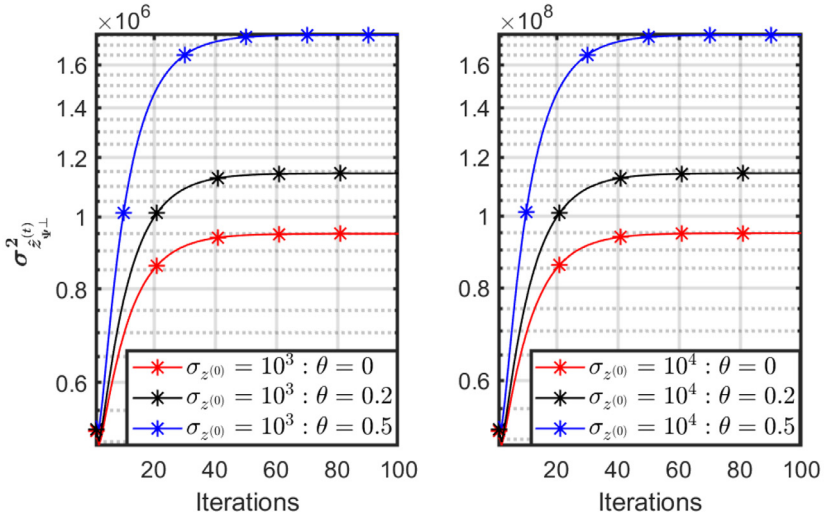
$$\begin{aligned} \min_{\boldsymbol{x}_i} \quad & \sum_{i\in\mathcal{N}} \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{Q}_i\boldsymbol{x}_i\|_2^2 \\ \text{s.t.} \quad & \forall (i,j) \in \mathcal{E} : \boldsymbol{x}_i = \boldsymbol{x}_j. \end{aligned} \tag{31}$$

In all experiments, we randomly draw the private data, i.e., $\boldsymbol{s}_i$ in the case of distributed average consensus and $\boldsymbol{Q}_i, \boldsymbol{y}_i$ in the case of distributed least squares, from a zero-mean Gaussian distribution with unit variance. In addition, we set $c = \gamma = 0.9$ and each entry of the auxiliary variable $\boldsymbol{z}^{(0)}$ is initialized with zero-mean Gaussian distributed noise having a variance $\sigma_{z^{(0)}}^2 = {\Delta^{(0)}}^2$, where $\Delta^{(0)}$ is the initial quantization cell-width. Moreover, for the proposed quantized approach, a one-bit (mid-rise) quantizer is used with cell-width $\Delta^{(t)}$, which means that we only transmit the signs of the $\boldsymbol{z}_{i|j}$s which will be reconstructed at the receiver by $\pm\Delta^{(t)}/2$.

In Fig. 3 we demonstrate experimental results to validate the conclusions drawn in Section 5.2.1. Figure 3 (a) shows that, in the absence of quantization, the variance of $\boldsymbol{z}_{\Psi^\perp}^{(t)}$ (we take the mean variance of all entries) in the case of $\theta = 0$ (PDMM) and $\theta = 0.5$ (ADMM) remains constant for all iterations. In addition, for $\theta = 0.2$, the variance decreases monotonically. Hence, for all $\theta \in [0, 1)$, the variance is lower bounded where the bound depends on the initialization of $\boldsymbol{z}^{(0)}$. As for the quantized case where $\hat{\boldsymbol{z}}^{(t)} = \boldsymbol{z}^{(0)} + \sum_{\tau=1}^{t}\hat{\boldsymbol{v}}^{(\tau)}$, from Fig. 3 (b) we can see that the variance first increases as $\Pi_{\Psi^\perp}\hat{\boldsymbol{v}}^{(t)} \neq 0$ and then converges as $\hat{\boldsymbol{v}}^{(t)}$ converges to zero. Similarly, the variance has a lower bound and it will be increased by increasing the initialization of $\boldsymbol{z}^{(0)}$. Overall, we conclude that the covariance of the auxiliary variable has a lower bound and it can be controlled by initialization. In what follows we will demonstrate the performance of the proposed approach in terms



(a) Variance of $\boldsymbol{z}^{(t)}_{\Psi^\perp}$ using non-quantized PDMM ($\theta = 0$) and generalized ADMM ($\theta = 0.2, 0.5$) under two different initializations: $\sigma_{z^{(0)}} = 10^3$ (left) and $\sigma_{z^{(0)}} = 10^4$ (right), respectively.

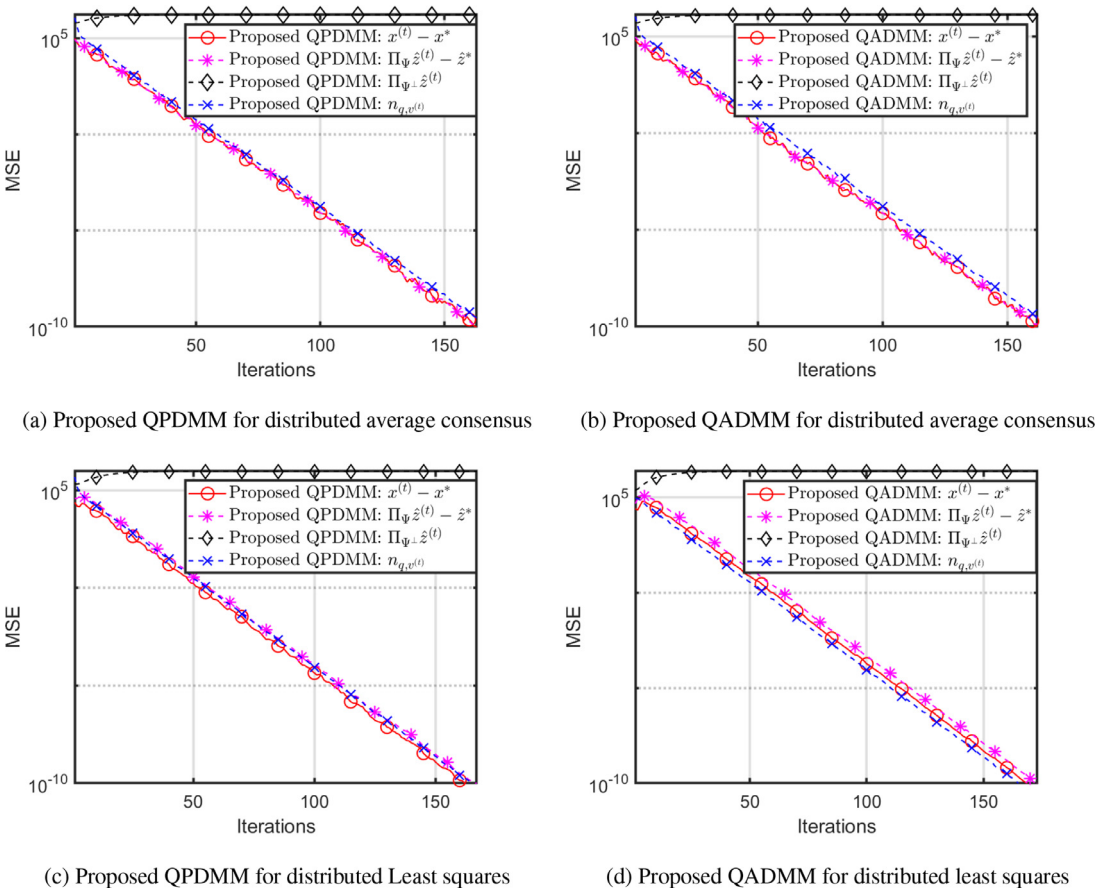


(b) Variance of $\hat{\boldsymbol{z}}^{(t)}_{\Psi^\perp}$ using the proposed quantized PDMM ($\theta = 0$) and generalized ADMM ($\theta = 0.2, 0.5$) under two different initializations: $\sigma_{z^{(0)}} = 10^3$ (left) and $\sigma_{z^{(0)}} = 10^4$ (right), respectively.

Fig. 3. Variance of $\boldsymbol{z}^{(t)}_{\Psi^\perp}$ and $\hat{\boldsymbol{z}}^{(t)}_{\Psi^\perp}$ using using the non-quantized (a) and the proposed quantized PDMM ($\theta = 0$) and generalized ADMM ($\theta = 0.2, 0.5$) for distributed average consensus application.

of the three requirements mentioned in Section 3.3. Without loss of generality, for ADMM we use $\theta = 0.5$.

1. **Output correctness**: From Fig. 4 we can see that applying the proposed adaptive differential quantization scheme to both PDMM (QPDMM) and ADMM (QADMM), the quantization noise $\boldsymbol{n}_{q,v^{(t)}}$ converges to zero and the optimization variable $\boldsymbol{x}^{(t)}$ converges to the optimal $\boldsymbol{x}^*$. These results validate the claim stated in Section 5.3: if the sequence $\{\boldsymbol{n}_{q,v^{(t)}}\}_{t>0}$ is finitely summable, the output correctness will be guaranteed. In addition, we also demonstrate the convergence behavior of $\hat{\boldsymbol{z}}^{(t)}$ through both projections into $\boldsymbol{\Pi}_\Psi$ and $\boldsymbol{\Pi}_{\Psi^\perp}$. Clearly, we can see that the convergence behavior of $\boldsymbol{\Pi}_{\Psi^\perp}\hat{\boldsymbol{z}}^{(t)}$ does not prevent $\boldsymbol{x}$ from converging to its optimum solution. Hence, the output correctness is not affected. Overall, we conclude that the proposed approach satisfies the output correctness requirement, i.e., accuracy is not compromised by considering both quantization and privacy. Additionally, it is generally applicable to both ADMM and PDMM.
2. **Communication cost**: Fig. 5 demonstrates the total communication cost (the amount of bits) of the proposed QADMM and QPDMM under three different privacy levels: $\sigma^2_{z^{(0)}} = 10^0, \sigma^2_{z^{(0)}} = 10^2, \sigma^2_{z^{(0)}} = 10^4$. Note that for the proposed approach we have $l = 1$ since a one-bit quantizer is used. We can see that the convergence rate of the proposed approach is invariant to the privacy level.
3. **Individual privacy**: Fig. 6 shows the individual privacy of an arbitrary honest node over iterations using the proposed approach under the condition that there is only one honest neighboring node when applied to the distributed average consensus problem. That is, the normalized mutual information measured based on (23) when $\mathcal{N}_{i,c} = d_i - 1$. We can see that the larger $\sigma^2_{z^{(0)}}$ is, the less individual privacy is revealed, i.e., the higher the privacy level is. Hence, the proposed approach is able to



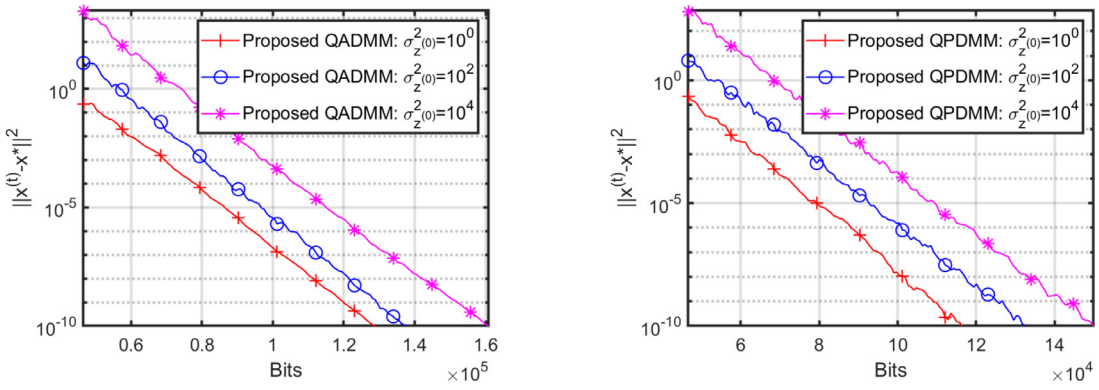
(a) Proposed QPDMM for distributed average consensus

(b) Proposed QADMM for distributed average consensus

(c) Proposed QPDMM for distributed Least squares

(d) Proposed QADMM for distributed least squares

Fig. 4. Convergence behavior (MSE) of different variables including $\boldsymbol{x}^{(t)} - \boldsymbol{x}^*$, $\Pi_\Psi \hat{\boldsymbol{z}}^{(t)} - \hat{\boldsymbol{z}}^*$, $\Pi_{\Psi^\perp} \hat{\boldsymbol{z}}^{(t)}$ and $\boldsymbol{n}_{q,v^{(t)}}$ in terms of iteration numbers using the proposed quantized PDMM and ADMM algorithm (QPDMM and QADMM, respectively) for the distributed average consensus and distributed least-squares applications when setting $\sigma_{z^{(0)}} = 10^3$.



(a) Proposed QADMM

(b) Proposed QPDMM

Fig. 5. MSE in terms of communication cost (the amount of bits) of the proposed QADMM and QPDMM for three different noise levels $\sigma^2_{z^{(0)}}$ of the auxiliary variables in distributed average consensus application.

guarantee individual privacy by controlling the variance of the initialized $\boldsymbol{z}^{(0)}$, i.e., $\sigma^2_{z^{(0)}}$.

6.2. Comparison with existing approaches

We now compare the performance of the proposed QADMM with existing privacy-preserving approaches including subspace perturbation (SP) based approach [20], secret sharing (SS) based approach [55] and differential privacy (DP) based approach [56]. To ensure a fair comparison, we insert the same amount of noise in all these algorithms. More specifically, all inserted noise are Gaussian distributed with zero mean and variance 10^2. Additionally, all algorithms are based on the ADMM optimizer. Note that none of the existing algorithms consider any quantization scheme but assume infinite precision, in the experiments we use the default MATLAB double precision floating-point format for simulations. That is, the number of bits to represent each massage is set to $l = 64$.

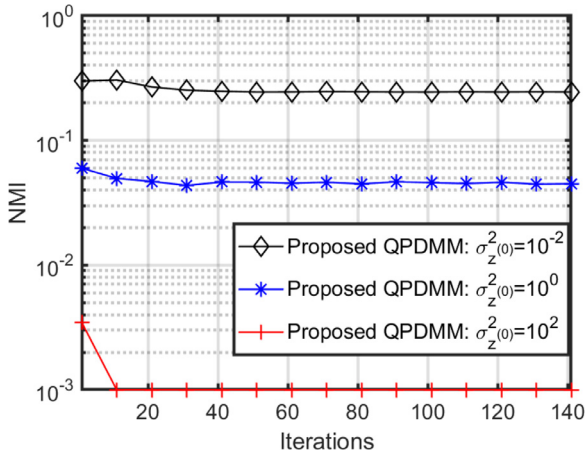


Fig. 6. Individual privacy (normalized mutual information (NMI)) of the proposed QPDMM in terms of iteration numbers for three different noise levels in distributed average consensus application.

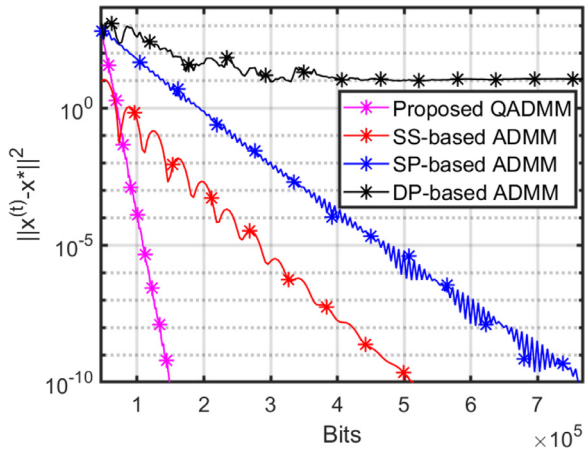


Fig. 7. Communication cost (bits) comparisons of the proposed QADMM algorithm and the existing subspace perturbation approach [20], secret sharing approach [55] and differential privacy approach [56] under the same privacy level in distributed average consensus application.

In Fig. 7, we demonstrate both the output correctness and communication cost performances of existing approaches and the proposed approach under the same amount of noise insertion.

1. **Communication cost**: As expected, the proposed algorithm significantly reduces the communication cost compared to all existing approaches. This is because the proposed algorithm requires only 1 bit for transmitting each message, while for the existing algorithms the cost is 64 bits as no quantization is considered.
2. **Output correctness**: We can see that all approaches output an accurate result except for the differential privacy based approach, i.e., it suffers from a privacy-accuracy trade-off. Hence, the output correctness of the proposed approach is not compromised by considering quantization and privacy. Overall, we conclude that, with the help of adaptive differential quantization, the proposed algorithm addresses the trade-off between privacy and communication cost, without sacrificing the accuracy.

A remark is in order here. Among all existing privacy-preserving approaches, i.e., subspace perturbation, secret sharing and differential privacy based approaches, the proposed approach is obtained by applying adaptive differential quantization to the subspace perturbation based approaches such that the communication cost is reduced without sacrificing the individual privacy and output correctness. For the other two types of approaches it would also be interesting to investigate how quantization affects their performances in terms of privacy and accuracy.

7. Conclusion

In this paper, we proposed a novel yet general communication efficient privacy-preserving distributed optimization approach using adaptive differential quantization. By adopting an adaptive quantizer that dynamically decreases its cell-width for each iteration to reduce the communication cost and making use of additive noise insertion to achieve privacy-preservation, we are able to address the trade-off between privacy and communication cost without compromising the algorithm accuracy. In addition, the proposed algorithm is able to protect privacy of any honest node against the passive adversary by requiring only one honest neighboring node. Moreover, the proposed method is computationally very lightweight in its way of dealing with an eavesdropping adversary as no secure encryption is needed, except for in the initialization step. Finally, numerical results were conducted, which confirm the desirable properties of the proposed approach in terms of accuracy, privacy and communication cost and show that the proposed approach has superior performance compared to the existing privacy-preserving approaches.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of Theorem 5.1

Proof. By inspection of (26) we conclude that

$$\begin{aligned}\boldsymbol{z}_{\Psi^\perp}^{(t)} &= (\theta \boldsymbol{I}_{2m} + (1-\theta)\boldsymbol{P})\boldsymbol{z}_{\Psi^\perp}^{(t-1)} \\ &= (\theta \boldsymbol{I}_{2m} + (1-\theta)\boldsymbol{P})^t \boldsymbol{z}_{\Psi^\perp}^{(0)}.\end{aligned}$$

Let $\boldsymbol{T} = \theta \boldsymbol{I}_{2m} + (1-\theta)\boldsymbol{P}$ so that $\boldsymbol{z}_{\Psi^\perp}^{(t)} = \boldsymbol{T}^t \boldsymbol{z}_{\Psi^\perp}^{(0)}$. The eigenvalues of $\boldsymbol{T}$ are found by finding the roots of the characteristic polynomial

$$\begin{aligned}p(\lambda) &= \det(\boldsymbol{T} - \lambda \boldsymbol{I}_{2m}) \\ &= \left((\theta-\lambda)^2 - (1-\theta)^2)\right)^m = (\lambda-1)^m(\lambda - 2\theta + 1)^m = 0,\end{aligned}$$

and we conclude that $\lambda_1 = 1$ and $\lambda_2 = 2\theta - 1$, both having algebraic and geometric multiplicity m. The corresponding eigenvectors $\boldsymbol{x}_i \in \mathbb{R}^{2m}$ are found by solving $(\boldsymbol{T} - \lambda_i \boldsymbol{I}_{2m})x = 0$ so that $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = (\boldsymbol{I}_m \boldsymbol{I}_m)^T$ corresponding to $\lambda_1 = 1$, and $(\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{2m}) = (\boldsymbol{I}_m - \boldsymbol{I}_m)^T$ corresponding to $\lambda_1 = 2\theta - 1$. With this, we can express $\boldsymbol{T}$ as

$$\boldsymbol{T} = \frac{1}{2}\begin{pmatrix}\boldsymbol{I}_m & \boldsymbol{I}_m \\ \boldsymbol{I}_m & -\boldsymbol{I}_m\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_m & \\ & (2\theta-1)\boldsymbol{I}_m\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_m & \boldsymbol{I}_m \\ \boldsymbol{I}_m & -\boldsymbol{I}_m\end{pmatrix},$$

and we conclude that

$$\begin{aligned}\boldsymbol{z}_{\Psi^\perp}^{(t)} = \boldsymbol{T}^t \boldsymbol{z}_{\Psi^\perp}^{(0)} &= \frac{1}{2}\begin{pmatrix}\boldsymbol{I}_m & \boldsymbol{I}_m \\ \boldsymbol{I}_m & -\boldsymbol{I}_m\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_m & \\ & (2\theta-1)\boldsymbol{I}_m\end{pmatrix}^k\begin{pmatrix}\boldsymbol{I}_m & \boldsymbol{I}_m \\ \boldsymbol{I}_m & -\boldsymbol{I}_m\end{pmatrix}\boldsymbol{z}_{\Psi^\perp}^{(0)} \\ &= \frac{1}{2}(\boldsymbol{I}_{2m} + \boldsymbol{P})\boldsymbol{z}_{\Psi^\perp}^{(0)} + \frac{1}{2}(2\theta-1)^t(\boldsymbol{I}_{2m} - \boldsymbol{P})\boldsymbol{z}_{\Psi^\perp}^{(0)},\end{aligned}$$

which completes the proof. □

CRediT authorship contribution statement

Qiongxiu Li: Conceptualization, Methodology, Software, Validation, Writing – original draft, Visualization. **Richard Heusdens:** Conceptualization, Resources, Writing – review & editing. **Mads Græsbøll Christensen:** Supervision, Writing – review & editing, Funding acquisition.

References

[1] M. Anderson, Technology Device Ownership, 2015, Pew Research Center, 2015.
[2] J. Poushter, Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies vol. 22 (2016) 1–44.
[3] A. Bertrand, S. Doclo, S. Gannot, N. Ono, T. Waterschoot, Special issue on wireless acoustic sensor networks and ad hoc microphone arrays, Signal Process. 107 (2015) 1–3.
[4] S. Modalavalasa, U.K. Sahoo, A.K. Sahoo, S. Baraha, A review of robust distributed estimation strategies over wireless sensor networks, Signal Process. 188 (2021) 108–150.
[5] S. Xu, R.C. de Lamare, H.V. Poor, Distributed low-rank adaptive estimation algorithms based on alternating optimization, Signal Process. 144 (2018) 41–51.
[6] Y. Zeng, R.C. Hendriks, Distributed estimation of the inverse of the correlation matrix for privacy preserving beamforming, Signal Process. 107 (2015) 109–122.
[7] Z. Huang, S. Mitra, N. Vaidya, Differentially private distributed optimization, in: Proceedings of the International Conference on Distributed Computing and Networking, 2015, pp. 1–10.
[8] S. Han, U. Topcu, G.J. Pappas, Differentially private distributed constrained optimization, IEEE Trans. Autom. Control. 62 (1) (2016) 50–64.
[9] E. Nozari, P. Tallapragada, J. Cortés, Differentially private distributed convex optimization via functional perturbation, IEEE Trans. Control Netw. Syst. 5 (1) (2018) 395–408.
[10] T. Zhang, Q. Zhu, Dynamic differential privacy for ADMM-based distributed classification learning, IEEE Trans. Inf. Forensics Secur. 12 (1) (2016) 172–187.
[11] X. Zhang, M.M. Khalili, M. Liu, Recycled ADMM: improve privacy and accuracy with less computation in distributed algorithms, in: Proceedings of the 56th Annual Allerton Conference on Communication, Control, and Computing, 2018, pp. 959–965.
[12] X. Zhang, M.M. Khalili, M. Liu, Improving the privacy and accuracy of ADMM-based distributed algorithms, in: Proceedings of the International Conference on Machine Learning, 2018, pp. 5796–5805.
[13] Y. Xiong, J. Xu, K. You, J. Liu, L. Wu, Privacy preserving distributed online optimization over unbalanced digraphs via subgradient rescaling, IEEE Trans. Control Netw. Syst. (2020).
[14] K. Tjell, R. Wisniewski, Privacy preservation in distributed optimization via dual decomposition and ADMM, in: Proceedings of the IEEE 58th Conference on Decision and Control, 2020, pp. 7203–7208.
[15] K. Tjell, I. Cascudo, R. Wisniewski, Privacy preserving recursive least squares solutions, in: Proceedings of the European Control Conference, 2019, pp. 3490–3495.
[16] R. Cramer, I.B. Damgård, J.B. Nielsen, Secure Multiparty Computation and Secret Sharing, Cambridge University Press, 2015.
[17] I. Damgård, V. Pastro, N. Smart, S. Zakarias, Multiparty computation from somewhat homomorphic encryption, in: Advances in Cryptology–CRYPTO, Springer, 2012, pp. 643–662.
[18] Q. Li, R. Heusdens, M.G. Christensen, Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2020, pp. 5895–5899.
[19] Q. Li, R. Heusdens, M.G. Christensen, Convex optimization-based privacy-preserving distributed least squares via subspace perturbation, in: Proceedings of the European Signal Processing Conference, 2021, pp. 2110–2114.
[20] Q. Li, R. Heusdens, M.G. Christensen, Privacy-preserving distributed optimization via subspace perturbation: a general framework, IEEE Trans. Signal Process. 68 (2020) 5983–5996.
[21] T.C. Aysal, M.J. Coates, M.G. Rabbat, Distributed average consensus with dithered quantization, IEEE Trans. Signal Process. 56 (10) (2008) 4905–4918.
[22] S. Kar, J.M.F. Moura, Distributed consensus algorithms in sensor networks: quantized data and random link failures, IEEE Trans. Signal Process. 58 (3) (2010) 1383–1400.
[23] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Gossip consensus algorithms via quantized communication, Automatica 46 (1) (2010) 70–80.
[24] S. Zhu, B. Chen, Quantized consensus by the ADMM: probabilistic versus deterministic quantizers, IEEE Trans. Signal Process. 64 (7) (2016) 1700–1713.
[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations Trends Mach. Learn. 3 (1) (2011) 1–22.
[26] G. Zhang, R. Heusdens, Bi-alternating direction method of multipliers over graphs, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 3571–3575.
[27] G. Zhang, R. Heusdens, Distributed optimization using the primal-dual method of multipliers, IEEE Trans. Signal Process. 1 (1) (2018) 173–187.
[28] T. Sherson, R. Heusdens, W.B. Kleijn, Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory, IEEE Trans. Signal Inf. Process. Netw. 5 (2) (2018) 334–347.
[29] E. Ryu, S.P. Boyd, Primer on monotone operator methods, Appl. Comput. Math. 15 (1) (2016) 3–43.
[30] G. Giaconi, D. Gndz, H.V. Poor, Privacy-aware smart metering: progress and challenges, IEEE Signal Process. Mag. 35 (6) (2018) 59–78.
[31] D. Bogdanov, S. Laur, J. Willemson, Sharemind: a framework for fast privacy-preserving computations, in: Proceedings of the 13th European Symposium on Research in Computer Security, 2008, pp. 192–206.
[32] Q. Li, M.G. Christensen, A privacy-preserving asynchronous averaging algorithm based on Shamir's secret sharing, in: Proceedings of the European Signal Processing Conference, 2019, pp. 1–5.
[33] T.M. Cover, J.A. Tomas, Elements of Information Theory, John Wiley & Sons, 2012.
[34] Q. Li, J.S. Gundersen, R. Heusdens, M.G. Christensen, Privacy-preserving distributed processing: metrics, bounds, and algorithms, IEEE Trans. Inf. Forensics Secur. 16 (2021) 2090–2103.
[35] M. Lopuhaä-Zwakenberg, B. Škorić N. Li, Information-theoretic metrics for local differential privacy protocols, arXiv preprint arXiv:1910.07826(2019).
[36] Q. Li, M. Coutino, G. Leus, M.G. Christensen, Privacy-preserving distributed graph filtering, in: Proceedings of the European Signal Processing Conference, 2021, pp. 2155–2159.
[37] S. Yagli, A. Dytso, H.V. Poor, Information-theoretic bounds on the generalization error and privacy leakage in federated learning, in: IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020, pp. 1–5.
[38] Y. Bu, S. Zou, V.V. Veeravalli, Tightening mutual information-based bounds on generalization error, IEEE J. Sel. Areas Info. Theory. 1 (1) (2020) 121–130.
[39] A. Pensia, V. Jog, P.L. Loh, Generalization error bounds for noisy, iterative algorithms, in: Proceedings of the IEEE International Symposium on Information Theory (ISIT), 2018, pp. 546–550.
[40] J. Negrea, M. Haghifam, G.K. Dziugaite, A. Khisti, D.M. Roy, Information-theoretic generalization bounds for SGLD via data-dependent estimates, in: Proceedings of the Neural Information Processing Systems (NeurIPS), 2019, pp. 11013–11023.
[41] P. Cuff, L. Yu, Differential privacy as a mutual information constraint, in: Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 43–54.
[42] M. Gtz, A. Machanavajjhala, G. Wang, X. Xiao, J. Gehrke, Publishing search logs–a comparative study of privacy guarantees, IEEE Trans. Knowl. Data. Eng. 24 (2011) 520–532.
[43] A. Haeberlen, B.C. Pierce, A. Narayan, Differential privacy under fire, in: Proceedings of the 20th USENIX Conference on Security, vol. 33, 2011.
[44] A. Korolova, K. Kenthapadi, N. Mishra, A. Ntoulas, Releasing search queries and clicks privately, in: Proceedings of the International Conference on World Wide Web, 2009, pp. 171–180.
[45] D. Kifer, A. Machanavajjhala, No free lunch in data privacy, in: SIGMOD, 2011, pp. 193–204.
[46] R.M. Gray, D.L. Neuhoff, Quantization, IEEE Trans. Inf. Theory 44 (6) (1998) 2325–2383.
[47] D.H.M. Schellekens, T. Sherson, R. Heusdens, Quantisation effects in PDMM: a first study for synchronous distributed averaging, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2017, pp. 4237–4241.
[48] J.A.G. Jonkman, T. Sherson, R. Heusdens, Quantisation effects in distributed optimisation, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 3649–3653.
[49] J. Liang, J. Fadili, G. Peyré, Convergence rates with inexact non-expansive operators, Math. Program. 159 (1–2) (2016) 403–434.
[50] D. Dolev, C. Dwork, O. Waarts, M. Yung, Perfectly secure message transmission, J. Assoc. Comput. Mach. 40 (1) (1993) 17–47.
[51] E.K. Ryu, S. Boyd, A primer on monotone operator methods, Appl. Comput. Math. 15 (1) (2016) 3–43.
[52] J. Dall, M. Christensen, Random geometric graphs, Phys. Rev. E 66 (1) (2002) 016121.
[53] Q. Li, I. Cascudo, M.G. Christensen, Privacy-preserving distributed average consensus based on additive secret sharing, in: Proceedings of the European Signal Processing Conference, 2019, pp. 1–5.
[54] N. Gupta, J. Katz, N. Chopra, Privacy in distributed average consensus, IFAC-PapersOnLine 50 (1) (2017) 9515–9520.
[55] N. Gupta, J. Kat, N. Chopra, Statistical privacy in distributed average consensus on bounded real inputs, in: ACC, 2019, pp. 1836–1841.
[56] E. Nozari, P. Tallapragada, J. Cortés, Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design, Automatica 81 (2017) 221–231.
[57] J. He, L. Cai, C. Zhao, P. Cheng, X. Guan, Privacy-preserving average consensus: privacy analysis and algorithm design, IEEE Trans. Signal Inf. Process. Netw. 5 (1) (2019) 127–138.
[58] M.H. Ruan, M. Ahmad, Y.Q. Wang, Secure and privacy-preserving average consensus, in: Proceedings of the Workshop on Cyber-Physical Systems Security and PrivaCy, 2017, pp. 123–129.
[59] Y. Mo, R.M. Murray, Privacy preserving average consensus, IEEE Trans. Automat. Control. 62 (2) (2017) 753–765.