2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Distributed Semi-Stochastic Optimization with Quantization Refinement

Neil McGlohon and Stacy Patterson

Abstract— **We consider the problem of regularized regression in a network of communication-constrained devices. Each node has local data and objectives, and the goal is for the nodes to optimize a global objective. We develop a distributed optimization algorithm that is based on recent work on semi-stochastic proximal gradient methods. Our algorithm employs iteratively refined quantization to limit message size. We present theoretical analysis and conditions for the algorithm to achieve a linear convergence rate. Finally, we demonstrate the performance of our algorithm through numerical simulations.**

I. Introduction

We consider the problem of distributed optimization in a network where communication is constrained, for example a wireless sensor network. In particular, we focus on problems where each node has local data and objectives, and the goal is for the nodes to learn a global objective that includes this local information. Such problems arise in networked systems problems such as estimation, prediction, resource allocation, and control.

Recent works have proposed distributed optimization methods that reduce communication by using quantization. For example, in [1], the authors propose a distributed algorithm to solve unconstrained problems based on a centralized inexact proximal gradient method [2]. In [3], the authors extend their work to constrained optimization problems. In these algorithms, the nodes compute a full gradient step in each iteration, requiring quantized communication between every pair of neighboring nodes. Quantization has been applied in distributed consensus algorithms [4], [5], [6] and distributed subgradient methods [7].

In this work, we address the specific problem of distributed regression with regularization over the variables across all nodes. Applications of our approach include distributed compressed sensing, LASSO, group LASSO, and regression with Elastic Net regularization, among others. Our approach is inspired by [1], [3]. We seek to further reduce per-iteration communication by using an approach based on a stochastic proximal gradient algorithm. This approach only requires communication between a small subset of nodes in each iteration. In general, stochastic gradients may suffer from slow convergence. Thus any per-iteration communication savings could be counter-acted by an extended number of iterations. Recently, however, several works have proposed *semi-stochastic* gradient methods [8], [9], [10]. To reduce the variance of the iterates generated by a stochastic approach, these algorithms periodically incorporate a full gradient computation. It has been shown that these algorithms achieve a linear rate of convergence to the optimal solution.

We propose a distributed algorithm for regularized regression based on the centralized semi-stochastic proximal gradient of [10]. In most iterations, only a subset of nodes need communicate. We further reduce communication overhead by employing quantized messaging. Our approach reduces both the length of messages sent between nodes as well as the number of messages sent in total to converge to the optimal solution. The detailed contributions of our work are as follows:

- We extend the centralized semi-stochastic proximal gradient algorithm to include errors in the gradient computations and show the convergence rate of this inexact algorithm.
- We propose a distributed optimization algorithm based on this centralized algorithm that uses iteratively refined quantization to limit message size.
- We show that our distributed algorithm is equivalent to the centralized algorithm, where the errors introduced by quantization can be interpreted as inexact gradient computations. We further design quantizers that guarantees a linear convergence rate to the optimal solution.
- We demonstrate the performance of the proposed algorithm in numerical simulations.

The remainder of this paper is organized as follows. In Section II, we present the centralized inexact proximal gradient algorithm and give background on quantization. In Section III, we give the system model and problem formulation. Section IV details our distributed algorithm. Section V provides theoretical analysis of our proposed algorithm. Section VI presents our simulation results, and we conclude in Section VII.

II. Preliminaries

A. Inexact Semi-Stochastic Proximal Gradient Algorithm

We consider an optimization problem over the form:

$$\underset{x \in \mathbb{R}^P}{\text{minimize}} \quad G(x) = F(x) + R(x), \tag{1}$$

where $F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$, and the following assumptions are satisfied.

Assumption 1: Each $f_i(x)$ is differentiable, and its gradient $\nabla f_i(x)$ is Lipschitz continuous with constant L_i, i.e., for all $x, y \in \mathbb{R}^P$,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|. \tag{2}$$

*This work was funded in part by NSF grants 1553340 and 1527287.

N. McGlohon and S. Patterson are with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. (mcglon@rpi.edu, sep@cs.rpi.edu)

978-1-4673-8682-1/$31.00 ©2016 AACC

Algorithm 1 Inexact Prox-SVRG.

```
**Initialize:** $\tilde{x}^{(s)} = 0$
**for** $s = 0, 1, 2, \dots$ **do**
  $\tilde{g}^{(s)} = \nabla F(\tilde{x}^{(s)})$
  $x^{(s_0)} = \tilde{x}^{(s)}$
  **for** $t = 0, 1, 2, \dots, T-1$ **do**
    Choose $\ell$ uniformly at random from $\{1, \dots, N\}$.
    $v^{(s_t)} = \nabla f_\ell(x^{(s_t)}) - \nabla f_\ell(\tilde{x}^{(s)}) + \tilde{g}^{(s)} + e^{(s_t)}$
    $x^{(s_{t+1})} = \text{prox}_{\eta R}(\tilde{x}^{(s_t)} - \eta v^{(s_t)})$
  **end for**
  $\tilde{x}^{(s+1)} = \frac{1}{T}\sum_{t=1}^{T} \tilde{x}^{(s_t)}$
**end for**
```

Assumption 2: The function $R(x)$ is lower semicontinuous, convex, and its effective domain, $\text{dom}(R) := \{x \in \mathbb{R}^P \mid R(x) < +\infty\}$, is closed.

Assumption 3: The function $G(x)$ is strongly convex with parameter $\mu > 0$, i.e., for all $x, y \in \text{dom}(R)$ and for all $\xi \in \partial G(x)$,

$$G(y) \geq G(x) + \xi^{\mathsf{T}}(y - x) + \tfrac{\mu}{2}\|y - x\|^2, \tag{3}$$

where $\partial G(x)$ is the subdifferential of G at x. This strong convexity may come from either $F(x)$ or $R(x)$ (or both).

Problem (1) can be solved using a stochastic proximal gradient algorithm [11] where, in each iteration, a single ∇f_ℓ is computed for a randomly chosen $\ell \in \{1, \dots, N\}$, and the iterate is updated accordingly as,

$$x^{(t+1)} = \text{prox}_{\eta R}(x^{(t)} - \eta^{(t)} \nabla f_\ell(x^{(t)})).$$

Here, $\text{prox}_{\eta R}(\cdot)$ is the proximal operator

$$\text{prox}_{\eta R}(v) = \operatorname*{arg\,min}_{y \in \mathbb{R}^p} \frac{1}{2}\|y - v\|^2 + \eta R(y).$$

While stochastic methods offer the benefit of reduced per-iteration computation over standard gradient methods, the iterates may have high variance. These methods typically use a decreasing step-size $\eta^{(t)}$ to compensate for this variance, resulting in slow convergence. Recently, Xiao and Zhang proposed a semi-stochastic proximal gradient algorithm, Prox-SVRG that reduces the variance by periodically incorporating a full gradient computation [10]. This modification allows Prox-SVRG to use a constant step size, and thus, Prox-SVRG achieves a linear convergence rate.

We extend Prox-SVRG to include a zero-mean error in the gradient computation. Our resulting algorithm, Inexact Prox-SVRG, is given in Algorithm 1. The algorithm consists of an outer loop where the full gradient is computed and an inner loop where the iterate is updated based on both the stochastic and full gradients.

The following theorem states the convergence behavior of Algorithm 1.

Theorem 1: Let $\{\tilde{x}^{(s)}\}_{s \geq 0}$ be the sequence generated by Algorithm 1, with $0 < \eta < \frac{1}{4\overline{L}}$, where $\overline{L} = \max_i L_i$. Assume that the functions R, G, and f_i, $i = 1, \dots, N$, satisfy Assumptions 1, 2, and 3, and that the errors $e^{(s_t)}$ are zero-mean and uncorrelated with the iterates $x^{(s_t)}$ and their gradients $\nabla f_i(x^{(s_t)})$. Let $x^\star = \arg\min_x G(x)$, and let T be such that,

$$\alpha = \frac{1}{\mu\eta(1 - 4\overline{L}\eta)T} + \frac{4\overline{L}\eta(T+1)}{(1 - 4\overline{L}\eta)T} < 1.$$

Then,

$$\begin{aligned}\mathbf{E}&\left[G(\tilde{x}^{(s)}) - G(x^\star)\right] \\ &\leq \alpha^s \left(G(\tilde{x}^{(0)}) - G(x^\star) + \beta \sum_{i=1}^{s} \alpha^{-i} \Gamma^{(i)} \right),\end{aligned}$$

where $\beta = \frac{\eta}{T(1 - 4\overline{L}\eta)}$ and $\Gamma^{(i)} = \sum_{t=0}^{T-1} \mathbf{E}\|e^{(i_t)}\|^2$.

The proof of this theorem is given in the technical report [12].

From this theorem, we can derive conditions for the algorithm to converge to the optimal $x^\star$. Let the sequence $\{\Gamma^{(s)}\}_{s \geq 0}$ decrease linearly at a rate κ. Then

1) If $\kappa < \alpha$, then $\mathbf{E}\left[G(\tilde{x}^{(s)}) - G(x^\star)\right]$ converges linearly with a rate of α.
2) If $\alpha < \kappa < 1$, then $\mathbf{E}\left[G(\tilde{x}^{(s)}) - G(x^\star)\right]$ converges linearly with a rate of κ.
3) If $\kappa = \alpha$, then $\mathbf{E}\left[G(\tilde{x}^{(s)}) - G(x^\star)\right]$ converges linearly with a rate in $O(s\alpha^s)$.

B. Subtractively Dithered Quantization

We employ a subtractively dithered quantizer to quantize values before transmission. We use a substractively dithered quantizer rather than non-subtractively dithered quantizer because the quantization error of the subtractively dithered quantizer is not correlated with its input. We briefly summarize the quantizer and its key properties below.

Let z be real number to be quantized into n bits. The quantizer is parameterized by an interval size U and a midpoint value $\overline{z} \in \mathbb{R}$. Thus the quantization interval is $[\overline{z} - U/2, \overline{z} + U/2]$, and the quantization step-size is $\Delta = \frac{U}{2^n - 1}$. We first define the uniform quantizer,

$$q(z) \triangleq \overline{z} + \text{sgn}(z - \overline{z}) \cdot \Delta \cdot \left\lfloor \frac{|z - \overline{z}|}{\Delta} + \frac{1}{2} \right\rfloor. \tag{4}$$

In subtractively dithered quantization, a dither ν is added to z, the resulting value is quantized using a uniform quantizer, and then transmitted. The recipient then subtracts ν from this value. The subtractively dithered quantized value of z, denoted $\hat{z}$, is thus

$$\hat{z} = Q(z) \triangleq q(z + \nu) - \nu. \tag{5}$$

Note that this quantizer requires both the sender and recipient to use the same value for ν, for example, by using the same pseudorandom number generator.

The following theorem describes the statistical properties of the quantization error.

Theorem 2 (See [13]): Let $z \in [\overline{z} - U/2, \overline{z} + U/2]$ and $\hat{z} = Q(z)$, for $Q(\cdot)$ in (5). Further, let ν is a real number drawn uniformly at random from the interval $(-\Delta/2, \Delta/2)$. The quantization error $\varepsilon(z) \triangleq z - \hat{z}$ satisfies the following:

1) $\mathbf{E}\left[\varepsilon(z)\right] = \mathbf{E}\left[\nu\right] = 0$.
2) $\mathbf{E}\left[\varepsilon(z)^2\right] = \mathbf{E}\left[\nu^2\right] = \frac{\Delta^2}{12}$.

3) $\mathbf{E}\left[z\varepsilon(z)\right] = \mathbf{E}\left[z\right]\mathbf{E}\left[\varepsilon(z)\right] = 0$.
4) For z_1 and z_2 in the interval $[\overline{z} - U/2, \overline{z} + U/2]$, $\mathbf{E}\left[\varepsilon(z_1)\varepsilon(z_2)\right] = \mathbf{E}\left[\varepsilon(z_1)\right]\mathbf{E}\left[\varepsilon(z_2)\right] = 0$.

With some abuse of notation, we also write $Q(v)$ where v is a vector. In this case, the quantization operator is applied to each component of v independently, using a vector-valued midpoint and the same scalar-valued interval bounds.

III. Problem Formulation

We consider a similar system model to that in [1]. The network is a connected graph of N nodes where inter-node communication is limited to the local neighborhood of each node. The neighbor set $\mathcal{N}_i$ consists of node i's neighbors and itself. The neighborhoods exist corresponding to the fixed undirected graph $G = (\mathcal{V}, \mathcal{E})$. We denote D as the maximum degree of the graph G.

Each node i has a state vector x_i with dimension m_i. The state of the system is $x = [x_1^\mathsf{T} x_2^\mathsf{T} \dots x_N^\mathsf{T}]^\mathsf{T}$. We let $x_{\mathcal{N}_i}$ be the vector consisting of the concatenation of states of all nodes in $\mathcal{N}_i$. For ease of exposition, we define the selecting matrices $\mathcal{A}_i$, $i = 1, \dots, N$, where $x_{\mathcal{N}_i} = \mathcal{A}_i x$ and the matrices $\mathcal{B}_{ij}$, $i, j = 1, \dots, N$ where $x_j = \mathcal{B}_{ij} x_{\mathcal{N}_i}$. These matrices each have ℓ_2-norm of 1.

Every node i has a local objective function over the states in $\mathcal{N}_i$. The distributed optimization problem is thus,

$$\underset{x \in \mathbb{R}^P}{\text{minimize}} \quad G(x) = F(x) + R(x), \tag{6}$$

where $F(x) = \frac{1}{N}\sum_{i=1}^N f_i(x_{\mathcal{N}_i})$. We assume that Assumptions 1, 2, and 3 are satisfied. Further, we require the following assumptions hold.

Assumption 4: For all i, $\nabla f_i(x_{\mathcal{N}_i})$ is linear or constant. This implies that, for a zero-mean random variable ν, $\mathbf{E}\left[\nabla f_i(x_{\mathcal{N}_i} + \nu)\right] = \nabla f_i(x_{\mathcal{N}_i})$.

Assumption 5: The proximal operation $\text{prox}_R(x)$ can be performed by each node locally, i.e.,

$$\text{prox}_R(x) = [\text{prox}_R(x_1)^\mathsf{T}\ \text{prox}_R(x_2)^\mathsf{T} \dots \text{prox}_R(x_N)^\mathsf{T}]^\mathsf{T}.$$

We note that Assumption 5 holds for standard regularization functions used in LASSO ($\|x\|_1$), group LASSO where each x_i is its own group, and Elastic Net regularization ($\lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|_2^2$).

In the next section, we present our distributed implementation of Prox-SVRG to solve Problem (6).

IV. Algorithm

Our distributed algorithm is given in Algorithm 2. In each outer iteration s, node i quantizes its iterate $\tilde{x}_i^{(s)}$ and the gradient $\nabla f_i^{(s)}$ and sends it to all of its neighbors. These values are quantized using two subtractively dithered quantizers, $Q_{a,i}^{(s)}$ and $Q_{b,i}^{(s)}$, whereby the sender (node i) sends an n bit representation and the recipient reconstructs the value from this representation and subtracts the dither. The midpoints for $Q_{a,i}^{(s)}$ and $Q_{b,i}^{(s)}$ are set to be the quantized values from the previous iteration. Thus, the recipients already know these midpoints. The quantized values (after the dither is subtracted) are denoted by $\hat{\tilde{x}}_i^{(s)}$ and $\hat{\nabla} f_i^{(s)}$, and the quantization errors are $a_i^{(s)}$ and $b_i^{(s)}$, respectively.

Algorithm 2 Inexact Semi-stochastic Gradient Descent as executed by node i

```
**Parameters:** inner loop size $T$, step size $\eta$
**Initialize:** $\tilde{x}_i^{(0)} = 0$, $\hat{\tilde{x}}_i^{(-1)} = 0$, $\hat{\nabla} f_i^{(-1)} = 0$
**for** $s = 0, 1, \dots$ **do**
  Update quantizer parameters:
    $U_{a,i}^{(s)} = C_a \kappa^{(s+1)/2}$, $\overline{x}_{a,i}^{(s)} = \hat{\tilde{x}}_i^{(s-1)}$,
    $U_{b,i}^{(s)} = C_b \kappa^{(s+1)/2}$, $\overline{\nabla} f_{b,i}^{(s)} = \hat{\nabla} f_i^{(s-1)}$
  Quantize local variable and send to all $j \in \mathcal{N}_i$:
    $\hat{\tilde{x}}_i^{(s)} = Q_{a,i}^{(s)}(\tilde{x}_i^{(s)}) = \tilde{x}_i^{(s)} + a_i^{(s)}$
  Compute: $\nabla f_i^{(s)} = \nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)})$
  Quantize gradient and send to all $j \in \mathcal{N}_i$:
    $\hat{\nabla} f_i^{(s)} = Q_{b,i}^{(s)}(\nabla f_i^{(s)}) = \nabla f_i^{(s)} + b_i^{(s)}$
  Compute: $\tilde{h}_i^{(s)} = \frac{1}{N}\sum_{j \in \mathcal{N}_i} \mathcal{B}_{ij} \hat{\nabla} f_j^{(s)}$
  Compute: $v_{ij}^{(s)} = -\mathcal{B}_{ij}\hat{\nabla} f_j^{(s)} + \tilde{h}_i^{(s)}$ for all $j \in \mathcal{N}_i$
  Update quantizer parameters:
    $U_{c,i}^{(s)} = C_c \kappa^{(s+1)/2}$, $\overline{x}_{c,i}^{(s)} = \hat{\tilde{x}}_i^{(s)}$,
    $U_{d,i}^{(s)} = C_d \kappa^{(s+1)/2}$, $\overline{\nabla} f_{d,i}^{(s)} = \hat{\nabla} f_i^{(s)}$
  $x_i^{(s_0)} = \tilde{x}_i^{(s)}$
  **for** $t = 0, 1, \dots, T-1$ **do**
    Randomly pick $\ell \in \{1, 2, 3, \dots, N\}$
    **if** $i \in \mathcal{N}_\ell$ **then**
      Quantize local variable and send to $\ell$:
        $\hat{x}_i^{(s_t)} = Q_{c,i}^{(s_t)}(x_i^{(s_t)}) = x_i^{(s_t)} + c_i^{(s_t)}$
      **if** $i = \ell$ **then**
        Compute: $\nabla f_i^{(s_t)} = \nabla f_i(\hat{x}_{\mathcal{N}_i}^{(s_t)})$
        Quantize gradient and send to all $j \in \mathcal{N}_i$:
          $\hat{\nabla} f_i^{(s_t)} = Q_{d,i}^{(s_t)}(\nabla f_i^{(s_t)}) = \nabla f_i^{(s_t)} + d_i^{(s_t)}$
      **end if**
      Update local variable:
        $x_i^{(s_{t+1})} = \text{prox}_{\eta R}(x_i^{(s_t)} - \eta(\mathcal{B}_{i\ell}\hat{\nabla} f_\ell^{(s_t)} + v_{i\ell}^{(s)}))$
    **else**
      Update local variable:
        $x_i^{(s_{t+1})} = \text{prox}_{\eta R}(x^{(s_t)} - \eta\tilde{h}_i^{(s)})$
    **end if**
  **end for**
  $\tilde{x}_i^{(s+1)} = \frac{1}{T}\sum_{t=1}^T x^{(s_t)}$
**end for**
```

For every iteration s of the outer loop of the algorithm, there is an inner loop of T iterations. In each inner iteration, a single node ℓ, chosen at random, computes its gradient. To do this, node ℓ and its neighbors exchange their states $x_i^{(s_t)}$ and gradients $\nabla f_i^{(s_t)}$. These values are quantized using two subtractively dithered quantizers, $Q_{c,i}^{(s_t)}$ and $Q_{d,i}^{(s_t)}$. The midpoints for these quantizers are $\hat{\tilde{x}}_i^{(s)}$ and $\hat{\nabla} f_i^{(s)}$. Each node sends these values to their neighbors before the inner loop, so all nodes are aware of the midpoints. The quantized values (after the dither is subtracted) are denoted by $\hat{x}^{(s_t)}$ and $\hat{\nabla} f_i^{(s_t)}$, and their quantization errors are $c_i^{(s_t)}$ and $d_i^{(s_t)}$, respectively. The quantization interval bounds $U_{a,i}^{(s)}$, $U_{b,i}^{(s)}$, $U_{c,i}^{(s)}$, and $U_{d,i}^{(s)}$, are initialized to C_a, C_b, C_c, and C_d, respectively, and each iteration, the bounds are multiplied by $\kappa^{1/2}$. Thus the quantizers are *refined* in each iteration.

The quantizers limit the length of a single variable transmission to n bits. In the outer loop of the algorithm, each node i sends its local variable, consisting of m_i quantized components, to every neighbor. It also sends its gradient, consisting of $|\mathcal{N}_i|m_i$ quantized components to every neighbor. Thus the number of bits exchanged by all nodes is $n\sum_{i=1}^{N}|\mathcal{N}_i|m_i + |\mathcal{N}_i|^2 m_i$ bits. In each inner iteration, only nodes $j \in \mathcal{N}_\ell$ exchange messages. Each node j quantizes m_j state variables and sends them to node ℓ. This yields a transmission of $n\sum_{j\in\mathcal{N}_\ell} m_j$ bits in total. In turn, node ℓ quantizes its gradient and sends it to all of its neighbors, which is $n|\mathcal{N}_\ell|^2 m_\ell$ total bits. Thus, in each inner iteration $n(|\mathcal{N}_\ell|^2 m_\ell + \sum_{j\in\mathcal{N}_\ell} m_j)$ bits are transmitted. The total number of bits transmitted in a single outer iteration is therefore,

$$n\left(\sum_{i=1}^{N}\left(|\mathcal{N}_i|m_i(1+|\mathcal{N}_i|)\right) + \sum_{t=0}^{T-1}\left(|\mathcal{N}_\ell|^2 m_\ell + \sum_{j\in\mathcal{N}_\ell} m_j\right)\right).$$

Let $D = \max_i |\mathcal{N}_i|$ and $\overline{m} = \max_i m_i$. An upper bound on the number bits transmitted by the algorithm in each outer iteration is $n\overline{m}(N+T)(D+D^2)$.

V. Algorithm Analysis

We now present our analysis of Algorithm 2. First we show that the algorithm is equivalent to Algorithm 1, where the quantization errors are encapsulated in the error term $e^{(s_t)}$. We also give an explicit expression for this error term.

Lemma 1: Algorithm 2 is equivalent to the Inexact Prox-SVG method in Algorithm 1, with

$$\begin{aligned} e^{(s_t)} &= \mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})\right) + \mathcal{A}_\ell^\mathsf{T} d_\ell^{(s_t)} \\ &- \mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) - \mathcal{A}_\ell^\mathsf{T} b_\ell^{(s)} \\ &+ \frac{1}{N}\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T}\left(\nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\right) + \frac{1}{N}\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T} b_i^{(s)}. \end{aligned}$$

Further, $\mathbf{E}\|e^{(s_t)}\|^2$ is upper-bounded by,

$$\begin{aligned} \mathbf{E}\|e^{(s_t)}\|^2 \leq 2\overline{L}^2 \sum_{j\in\mathcal{N}_\ell}\mathbf{E}\|c_j^{(s_t)}\|^2 + 2\overline{L}^2 \sum_{j\in\mathcal{N}_\ell}\mathbf{E}\|a_j^{(s)}\|^2 \\ + \mathbf{E}\|d_\ell^{(s_t)}\|^2 + 2\mathbf{E}\|b_\ell^{(s)}\|^2 + \frac{2}{N^2}\sum_{i=1}^{N}\mathbf{E}\|b_i^{(s)}\|^2. \end{aligned}$$

Proof: The error $e^{(s_t)}$ is:

$$\begin{aligned} e^{(s_t)} &= \mathcal{A}_\ell{}^\mathsf{T}\hat{\nabla} f_\ell^{(s_t)} - \mathcal{A}_\ell^\mathsf{T}\hat{\nabla} f_\ell^{(s)} + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i{}^\mathsf{T}\hat{\nabla} f_i^{(s)} \\ &\quad - \Big(\mathcal{A}_\ell{}^\mathsf{T}\nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)}) - \mathcal{A}_\ell{}^\mathsf{T}\nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)}) \\ &\qquad\qquad + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i{}^\mathsf{T}\nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\Big) \\ &= \mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})\right) + \mathcal{A}_\ell^\mathsf{T} d_\ell^{(s_t)} \\ &- \mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) - \mathcal{A}_\ell^\mathsf{T} b_\ell^{(s)} \\ &+ \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T}\left(\nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\right) + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T} b_i^{(s)}. \end{aligned}$$

We note that all quantization errors are zero-mean. Further, by Assumption 5, $\mathbf{E}\left[\nabla f_i(x+\delta)\right] = \nabla f_i(x)$, for a zero-mean random variable δ. Therefore, $\mathbf{E}\left[e^{(s_t)}\right] = 0$.

We now show that $e^{(s_t)}$ is is uncorrelated with $x^{(s_t)}$ and the gradients $\nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})$, $\ell = 1,\ldots,N$. Clearly, $x^{(s_t)}$ and $\nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})$ are uncorrelated with the terms of $e^{(s_t)}$ containing $d_\ell^{(s_t)}$, $b_\ell^{(s)}$, and $b_i^{(s)}$. In accordance with Assumption 5, the gradients ∇f_ℓ and ∇f_i are either linear or constant. If they are constant, then $\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)}) = 0$ and $\nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)}) = 0$. Thus, the terms in $e^{(s_t)}$ containing these differences are also 0. If they are linear, e.g., $\nabla f_\ell(z) = Hz + h$, for an appropriately sized, matrix H and vector h (possibly 0). Then,

$$\begin{aligned} &\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)}) \\ &\qquad = (H(x_{\mathcal{N}_\ell}^{(s_t)} + c_i^{(s_t)}) + h) - (Hx^{(s_t)} + h) = Hc_i^{(s_t)}. \end{aligned}$$

By Theorem 2, $c_i^{(s_t)}$ is uncorrelated with $x^{(s_t)}$. It is clearly also uncorrelated with $\nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})$. Similar arguments can be used to show that $x^{(s_t)}$ and $\nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})$ are uncorrelated with the remaining terms in $e^{(s_t)}$.

With respect to $\mathbf{E}\|e^{(s_t)}\|^2$, we have

$$\begin{aligned} \mathbf{E}\|e^{(s_t)}\|^2 = \mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}&\left(\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})\right) \\ &- \mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) \\ &+ \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T}\left(\nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\right)\|^2 \\ + \mathbf{E}\|\mathcal{A}_\ell^\mathsf{T} d_\ell^{(s_t)} &+ \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T} b_i^{(s)} - \mathcal{A}_\ell^\mathsf{T} b_\ell^{(s)}\|^2. \end{aligned}$$

The first term on the right hand side can be bounded using the fact that $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, as

$$\begin{aligned} \leq 2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}&\left(\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})\right)\|^2 \\ &+ 2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) \\ &\qquad + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T}\left(\nabla f_i(\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\right)\|^2. \end{aligned}$$

We now bound the first term in this expression,

$$\begin{aligned} &2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{x}_{\mathcal{N}_\ell}^{(s_t)}) - \nabla f_\ell(x_{\mathcal{N}_\ell}^{(s_t)})\right)\|^2 \\ &\qquad \leq 2\mathbf{E}(L_i^2\|\hat{x}_{\mathcal{N}_\ell}^{(s_t)} - x_{\mathcal{N}_\ell}^{(s_t)}\|^2) \leq 2\overline{L}^2\sum_{j\in\mathcal{N}_\ell}\mathbf{E}\|c_j^{(s_t)}\|^2, \end{aligned}$$

where the first inequality follows from Assumptions 1 and 5 and the fact that $\|\mathcal{A}_\ell\| = 1$. The second inequality follows from the independence of quantization errors (Theorem 2). Next we bound the second term,

$$\begin{aligned} &2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) \\ &\quad + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathcal{A}_i^\mathsf{T}\left(\nabla f_i(,\hat{\tilde{x}}_{\mathcal{N}_i}^{(s)}) - \nabla f_i(\tilde{x}_{\mathcal{N}_i}^{(s)})\right)\|^2 \\ &= 2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right) \\ &\qquad - \mathbf{E}\left[\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right)\right]\|^2 \\ &\leq 2\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T}\left(\nabla f_\ell(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)}) - \nabla f_\ell(\tilde{x}_{\mathcal{N}_\ell}^{(s)})\right)\|^2 \\ &\leq 2\mathbf{E}(L_i^2\|(\hat{\tilde{x}}_{\mathcal{N}_\ell}^{(s)} - \tilde{x}_{\mathcal{N}_\ell}^{(s)}\|^2) \\ &\leq 2\overline{L}^2\textstyle\sum_{j\in\mathcal{N}_\ell}\mathbf{E}\|a_j^{(s)}\|^2, \end{aligned}$$

where the first inequality uses the fact that for a random variable v, $\mathbf{E}\|v - \mathbf{E}v\|^2 = \mathbf{E}\|v\|^2 - \|\mathbf{E}v\|^2 \leq \mathbf{E}\|v\|^2$. The remaining inequalities follow from Assumptions 1 and 5, the fact that $\|\mathcal{A}_\ell\| = 1$, and the independence of the quantization errors.

Finally, again from the independence of the quantization errors, we have,

$$\begin{aligned} &\mathbf{E}\|\mathcal{A}_\ell^\mathsf{T} d_\ell^{(s_t)} + \tfrac{1}{N}\textstyle\sum_{i=1}^N \mathcal{A}_i^\mathsf{T} b_i^{(s)} - \mathcal{A}_\ell^\mathsf{T} b_\ell^{(s)}\|^2 \\ &\quad \leq \mathbf{E}\|\mathcal{A}_\ell^\mathsf{T} d_\ell^{(s_t)}\|^2 + \mathbf{E}\|\tfrac{1}{N}\textstyle\sum_{i=1}^N \mathcal{A}_i^\mathsf{T} b_i^{(s)}\|^2 - \mathcal{A}_\ell^\mathsf{T} b_\ell^{(s)}\|^2 \\ &\quad \leq \mathbf{E}\|d_\ell^{(s_t)}\|^2 + 2\mathbf{E}\|b_\ell^{(s)}\|^2 + \tfrac{2}{N^2}\textstyle\sum_{i=1}^N \mathbf{E}\|b_i^{(s)}\|^2. \end{aligned}$$

Combining these bounds, we obtain the desired result,

$$\begin{aligned} \mathbf{E}\|e^{(s_t)}\|^2 \leq 2\overline{L}^2 \sum_{j\in\mathcal{N}_\ell} \mathbf{E}\|c_j^{(s_t)}\|^2 + 2\overline{L}^2 \sum_{j\in\mathcal{N}_\ell} \mathbf{E}\|a_j^{(s)}\|^2 \\ + \mathbf{E}\|d_\ell^{(s_t)}\|^2 + 2\mathbf{E}\|b_\ell^{(s)}\|^2 + \frac{2}{N^2}\sum_{i=1}^N \mathbf{E}\|b_i^{(s)}\|^2. \end{aligned}$$

■

We next show that, if all of the values fall within their respective quantization intervals, then the error term $\Gamma^{(s)}$ decreases linearly with rate κ, and thus the algorithm converges to the optimal solution linearly with rate κ.

Theorem 3: Given p, if for all $1 \leq s \leq (p-1)$, the values of $\tilde{x}_i^{(s)}$, $\nabla f_i^{(s)}$, $x^{(s_t)}$, and $\nabla f_i^{(s_t)}$ fall inside of the respective quantization intervals $Q_{a,i}^{(s)}$, $Q_{b,i}^{(s)}$, $Q_{c,i}^{(s_t)}$, and $Q_{d,i}^{(s_t)}$, then $\Gamma^{(k)} \leq C\kappa^k$, where,

$$C = \frac{DT\overline{m}}{12(2^\ell - 1)^2}\left(2\overline{L}^2(C_a + C_b) + 2(\tfrac{N+1}{N})C_b + C_d\right),$$

with $D = \max_i |\mathcal{N}_i|$ and $\overline{m} = \max_i m_i$.

It follows that, for $\alpha < \kappa < 1$,

$$\begin{aligned} &\mathbf{E}\left[G(\tilde{x}^{(s)}) - G(x^\star)\right] \\ &\quad \leq \kappa^s \left(G(\tilde{x}^{(0)}) - G(x^\star) + \beta C\left(\frac{1}{1 - \frac{\alpha}{\kappa}}\right)\right). \end{aligned}$$

Proof: First we note that, by Theorem 2 and the update rule for the quantization intervals, we have:

$$\mathbf{E}\|a_i^{(s)}\|^2 \leq \tfrac{\overline{m}}{12}\left(\tfrac{U_{a,i}^{(s)}}{2^\ell-1}\right)^2 \leq \tfrac{\overline{m}}{12(2^\ell-1)^2}C_a\kappa^s$$

$$\mathbf{E}\|b_i^{(s)}\|^2 \leq \tfrac{D\overline{m}}{12}\left(\tfrac{U_{b,i}^{(s)}}{2^\ell-1}\right)^2 \leq \tfrac{D\overline{m}}{12(2^\ell-1)^2}C_b\kappa^s$$

$$\mathbf{E}\|c_i^{(s_t)}\|^2 \leq \tfrac{\overline{m}}{12}\left(\tfrac{U_{c,i}^{(s)}}{2^\ell-1}\right)^2 \leq \tfrac{\overline{m}}{12(2^\ell-1)^2}C_c\kappa^s$$

$$\mathbf{E}\|d_i^{(s_t)}\|^2 \leq \tfrac{D\overline{m}}{12}\left(\tfrac{U_{d,i}^{(s)}}{2^\ell-1}\right)^2 \leq \tfrac{D\overline{m}}{12(2^\ell-1)^2}C_d\kappa^s.$$

We use these inequalities to bound $\|e^{(s_t)}\|^2$,

$$\begin{aligned} &\mathbf{E}\|e^{(s_t)}\|^2 \leq 2\overline{L}^2 D\left(\tfrac{\overline{m}}{12(2^\ell-1)^2}C_c\kappa^s\right) \\ &+ 2\left(\overline{L}^2 D\tfrac{\overline{m}}{12(2^\ell-1)^2}C_a\kappa^s\right) + \tfrac{D\overline{m}}{12(2^\ell-1)^2}C_d\kappa^s \\ &+ 2\left(\tfrac{D\overline{m}}{12(2^\ell-1)^2}C_b\kappa^s\right) + \tfrac{2}{N}\left(\tfrac{D\overline{m}}{12(2^\ell-1)^2}C_b\kappa^s\right) \\ &= \tfrac{D\overline{m}}{12(2^\ell-1)^2}\left(2\overline{L}^2(C_a + C_c) + 2(\tfrac{N+1}{N})C_b + C_d\right)\kappa^s. \end{aligned}$$

Summing over $t = 0, \ldots, T-1$, we obtain,

$$\Gamma^{(s)} = \sum_{t=0}^{T-1} \mathbf{E}\|e^{(s_t)}\|^2 \leq C\kappa^s,$$

where

$$C = \tfrac{DT\overline{m}}{12(2^\ell-1)^2}\left(2\overline{L}^2(C_a + C_c) + 2(\tfrac{N+1}{N})C_b + C_d\right).$$

Applying Theorem 1, with $\kappa > \alpha$, we have,

$$\begin{aligned} &\mathbf{E}\left[G(\tilde{x}^{(s)}) - G(x^\star)\right] \\ &\quad \leq \alpha^s\left(G(\tilde{x}^{(0)}) - G(x^\star)\right) + \beta\sum_{i=1}^{s}\alpha^{s-i}C\kappa^i \\ &\quad \leq \kappa^s\left(G(\tilde{x}^{(0)}) - G(x^\star) + C\beta\sum_{i=1}^{s}\kappa^{-(s-i)}\alpha^{s-i}\right) \\ &\quad \leq \kappa^s\left(G(\tilde{x}^{(0)}) - G(x^\star) + C\beta\tfrac{1-(\frac{\alpha}{\kappa})^s}{1-\frac{\alpha}{\kappa}}\right) \\ &\quad \leq \kappa^s\left(G(\tilde{x}^{(0)}) - G(x^\star) + C\beta\left(\tfrac{1}{1-\frac{\alpha}{\kappa}}\right)\right). \end{aligned}$$

■

While we do not yet have theoretical guarantees that all values will fall within their quantization intervals, our simulations indicate that is always possible to find parameters C_a, C_b, C_c, and C_d, for which all values lie within their quantization intervals for all iterations. Thus, in practice, our algorithm achieves a linear convergence rate. We anticipate that it is possible to develop a programmatic approach, similar to that in [1], to identify values for C_a, C_b, C_c, and C_d that guarantee linear convergence. This is a subject of current work.

VI. Numerical Example

This section illustrates the performance of Algorithm 2 by solving a distributed linear regression problem with elastic net regularization.

We randomly generate a d-regular graph with $N = 40$ and uniform degree of 8, i.e., $\forall i\ |\mathcal{N}_i| = 9$. We set each subsystem size, m_i, to be 10. Each node has a local function $f_i(x_{\mathcal{N}_i}) = \|H_i x_{\mathcal{N}_i} - h_i\|^2$ where H_i is a 80×90 random matrix. We generate h_i by first generating a random vector x and then computing $h_i = H_i x$. The global objective function is:

$$G(x) = \frac{1}{N}\sum_{i}^{N} f_i(x_{\mathcal{N}_i}) + \lambda_1\|x\|_2 + \frac{\lambda_2}{2}\|x\|_1.$$

This simulation was implemented in Matlab and the optimal value $x^\star$ was computed using CVX. We set the total number of inner iterations to be $T = 2N$ and use the step size $\eta = 0.1/\overline{L}$. With these values, $\alpha < 1$, as required by Theorem 1. We set $\kappa = 0.97$, which ensures that $\kappa > \alpha$. We use the quantization parameters $C_a = 50,\ C_b = 300,\ C_c = 50,\ C_d = 400$. With these parameters, the algorithms values always fell within their quantization intervals.

Fig. 1 shows the performance of the algorithm where the number of bits n is 11, 13, and 15, as well as the performance of the algorithm without quantization. In these results, $x^{(s)}$

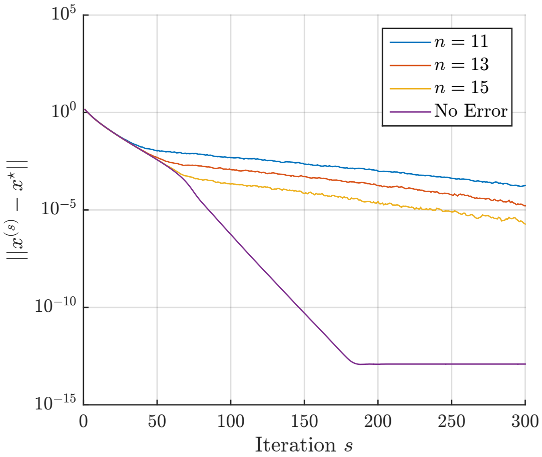

Fig. 1: Comparison of the performance of Algorithm 2 with differing quantized message lengths and that with no quantization applied.

is the concatenation of the $\tilde{x}_i^{(s)}$ vectors, for $i = 1, \ldots, N$. It is important to note the rate of convergence of the algorithm in all four cases is linear, and, performance improves as the number of bits increases.

VII. Conclusion

We have presented a distributed algorithm for regularized regression in communication-constrained networks. This algorithm is based on recently proposed semi-stochastic proximal gradient methods. Our algorithm reduces communication requirements by (1) using a stochastic approach where only a subset of nodes communicate in each iteration and (2) quantizing all messages. We have shown that this distributed algorithm is equivalent to a centralized version with inexact gradient computations, and we have used this equivalence to analyze the convergence rate of the distributed method. Finally, we have demonstrated the performance of our algorithm in numerical simulations.

In future work, we plan to extend our theoretical analysis to develop a programmatic way to identify initial quantization intervals. We also plan to explore the integration of more complex regularization functions.

References

[1] Y. Pu, M. N. Zeilinger, and C. N. Jones, “Quantization design for unconstrained distributed optimization,” in *Proc. American Control Conference*, 2015, pp. 1229–1234.

[2] M. Schmidt, N. L. Roux, and F. R. Bach, “Convergence rates of inexact proximal-gradient methods for convex optimization,” in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1458–1466.

[3] Y. Pu, M. N. Zeilinger, and C. N. Jones, “Quantization design for distributed optimization,” *arXiv:1504.02317*, 2015.

[4] A. Kashyap, T. Başar, and R. Srikant, “Quantized consensus,” *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[5] D. Thanou, E. Kokiopoulou, Y. Pu, and P. Frossard, “Distributed average consensus with quantization refinement,” *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 194–205, 2013.

[6] R. Carli, F. Fagnani, P. Frasca, T. Taylor, and S. Zampieri, “Average consensus on networks with transmission noise or quantization,” in *Proc. European Control Conference*, 2007, pp. 1852–1857.

[7] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, “Distributed subgradient methods and quantization effects,” in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 4177–4184.

[8] A. Nitanda, “Stochastic proximal gradient descent with acceleration techniques,” in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 1574–1582.

[9] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[10] L. Xiao and T. Zhang, “A proximal stochastic gradient method with progressive variance reduction,” *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[11] J. Duchi and Y. Singer, “Efficient online and batch learning using forward backward splitting,” *J. Mach. Learn. Res.*, vol. 10, pp. 2873–2898, 2009.

[12] N. McGlohon and S. Patterson, “Quantization design for distributed optimization,” *arXiv:1603.06306*, 2016.

[13] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, “Quantization and dither: A theoretical survey,” *Journal of the Audio Engineering Society*, vol. 40, no. 5, pp. 355–375, 1992.