ASYNCHRONOUS SUBGRADIENT OPTIMIZATION IN NOISY NETWORKS

De Wen Soh[‡] *Tony Q.S. Quek*[⋆] *Wee Peng Tay*[†]

[‡] Yale University, Department of Electrical Engineering, New Haven, CT 06520
[⋆]Institute for Infocomm Research, A*STAR, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632
[†] Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
dewen.soh@yale.edu, qsquek@i2r.a-star.edu.sg, wptay@ntu.edu.sg

ABSTRACT

We consider a gossip type of subgradient optimization that can be applied to noisy networks, where the communication links between nodes are noisy. Each node in the network has a function that is not known to all the other nodes and the goal is to cooperatively minimize the sum of all the functions in the network. Under noisy environment, we show that with our distributed optimization algorithm, the disagreements between the value of the nodes converge towards zero. Furthermore, for convex node function, we show that all values converge to the global optimal solution of the optimization problem.

***Index Terms*—** gossip, distributed optimization, subgradient, Lasso

1. INTRODUCTION

As the size and complexity of network increases, the notion of a central fusion center to manage the entire network becomes less feasible. As a result, research in distributed network algorithms have attracted the attention of many researchers in recent years. One group of these algorithms is known as gossip algorithms [1, 2], where nodes in a network communicate with one another to achieve a common goal by exchanging small amount of information.

Gossip algorithms have been used for many purposes, such as calculating the average of node values, and even for something as simple as disseminating information throughout an entire network. The application of gossip that we are interested is distributed subgradient optimization [3–9]. In this setting, each node in the system knows a function which all the other nodes do not know, and the nodes work cooperatively to minimize the sum of these functions. While previous deal with stochastic error that occurs from the distributed subgradient optimization, they do not deal with the presence of communication noise. When one node communicates with another node, the value that it receives from that node can be affected by noise. Such error noise can be problematic since it affects the accuracy of the iterations.

In this paper, we will investigate the distributed subgradient optimization of a sum of functions in the presence of communication noise. The communication noise occurs when nodes extract values from their neighbors. We study the convergence of the gossip based subgradient optimization method in noisy networks. Our main contributions are as follows:

(a) Our results state that in the presence of communication noise, the subgradient algorithm still causes the iterates of the nodes to reach a consensus. Therefore, the disagreements between the iterates converge towards zero. Specifically, for convex node functions, the node values converge towards the global solution with our distributed algorithm.

(b) We apply our algorithm to solve a distributed least-absolute shrinkage and selection operator (Lasso) problem under noisy conditions.

The rest of the paper is organized as follows. In Section 2, we present the system model and list down some preliminaries necessary for the rest of the paper. In Section 3, we state the convergence results for our distributed optimization algorithm in noisy networks. In Section 4, we present a numerical example in the setting of a distributed Lasso problem. Finally, we summarize and provide some concluding remarks in Section 5.

2. SYSTEM MODEL AND NOTATION

In a network of n nodes, let the nodes be labelled $1, 2, \cdots, n$ in some arbitrary order. Each of these nodes can communicate with its neighbours. We assume communication to be bidirectional, i.e. node i can communicate with node j if and only if node j can communicate with node i. Suppose each node has knowledge of a function $f_i : \mathbb{R} \to \mathbb{R}$, and the functions at the other nodes are never made known to itself. It can calculate the subgradient ∇f_i of the function f_i. Each node also has knowledge of an initial real number. These real numbers can be expressed in an n-dimensional vector $\mathbf{x}$, with $\mathbf{x}_i$ be the initial value at node i. Each node will pick a neighbour

to communicate with at Poisson rate 1 uniformly amongst its neighbours. In such a setting, the only information that gets transferred or shared is the real value each node has. The objective is to solve for the minimum value for the sum of all the functions of the network's nodes. Mathematically, this can be described as an optimization problem wrtiten as follows:

$$\begin{aligned} &\text{minimize} \quad f(x) := \sum_{i=1}^{m} f_i(x) \\ &\text{subject to } \ x \in \mathbb{R}. \end{aligned} \tag{1}$$

To define time, let $t(1)$ denote the first time any node decides to communicate with a neighbor, and correspondingly, let $t(k)$ be the k-th time that any nodes decides to communicate. Thus at $t(k)$, k communications have taken place. Let $s(k)$ denote the node that starts the communication, and let $r(k)$ be the node that node $s(k)$ decides to communicate with. Let $\mathbf{x}(k)$ denote the iterates of the nodes at time $t(k)$, with $\mathbf{x}(0) = \mathbf{x}$, and let $\mathbf{x}_i(k)$ denote the value of node i after communication at time $t(k)$. At time $t(k)$, node $s(k)$ will communicate with node $r(k)$ and node $s(k)$ will retrieve the value of node $r(k)$, and compute the average of these two values. Node $r(k)$ will do the same as well. We define

$$\bar{x}_{s(k),r(k)} = \frac{\mathbf{x}_{s(k)}(k-1) + \mathbf{x}_{r(k)}(k-1)}{2}.$$

Since the retrieval of values is subjected to communication error, we define a perturbation matrix $\mathbf{B}(k)$ and a perturbation vector $\mathbf{m}(k)$ to describe the communication noise. Let $(\mathbf{B}(k)\mathbf{m}(k))_i$ denote the i entry of $\mathbf{B}(k)\mathbf{m}(k)$. Because of the communication noise, the final computed average at node $s(k)$ becomes

$$\chi_i(k) = \bar{x}_{s(k),r(k)} + (\mathbf{B}(k)\mathbf{m}(k))_i \tag{2}$$

where $(\mathbf{B}(k)\mathbf{m}(k))_i$ is the difference between the computed average at node $s(k)$ and the actual average $\bar{x}_{s(k),r(k)}$.

If $i \notin \{s(k), r(k)\}$, then $\mathbf{x}_i(k) = \mathbf{x}_i(k-1)$. If $i \in \{s(k), r(k)\}$, then node i is updated and we have

$$\mathbf{x}_i(k) = \zeta_i(k) - \frac{1}{\Gamma_i(k)}(\nabla f_i(\zeta_i(k)) + \epsilon_i(k)) \tag{3}$$

where $\Gamma_i(k)$ is the number of updates of node i up to time $t(k)$ and $\epsilon_i(k)$ is the stochastic error associated with $\nabla f_i(\zeta_i(k))$.

Now, let $\mathbf{e}_i$ denotes the vector with value 1 for its i-th entry and 0 for the rest of its entries and I be the identity matrix. We define

$$\mathbf{W}(k) = I - \frac{1}{2}(\mathbf{e}_{s(k)} - \mathbf{e}_{r(k)})(\mathbf{e}_{s(k)} - \mathbf{e}_{r(k)})^T \tag{4}$$

and we can now express the update in (3) as follows:

$$\mathbf{x}(k) = \mathbf{W}(k)\mathbf{x}(k-1) + \mathbf{p}(k) + \mathbf{B}(k)\mathbf{m}(k) \tag{5}$$

where

$$\mathbf{p}(k) = -\sum_{i \in s(k), r(k)} \frac{1}{\Gamma_i(k)}(\nabla f_i(\chi_i(k)) + \epsilon_i(k))\mathbf{e_i}.$$

Let $y(k)$ be the average of the elements of $\mathbf{x}(k)$ and $\lambda_i\{\cdot\}$ be the ith largest eigenvalue of its argument. Using these notations, we can go on to describe some assumptions about our model.

2.1. Assumptions

These are the assumptions that we make about our model.

Assumption 1 *The gradients of the functions f_i are uniformly bounded, that is,*

$$\sup_{x \in \mathbb{R}, 1 \le i \le m} |\nabla f_i(x)| \le C,$$

for some constant $C > 0$.

Let $F(k-1)$ be the σ-algebra that is generated by the entire algorithm history up until time $t(k)$. For convenience, we denote $F(k-1)$ by F.

Assumption 2 *With probability 1, we have:*

(a) $\mathbb{E}[|\epsilon_i(k)|^2 \mid F] \le \nu^2$ *for all k and $i \in \mathcal{V}$, and for some $\nu \ge 0$.*

(b) $\mathbb{E}[\epsilon_{s(k)}(k) \mid F] = 0$ *and* $\mathbb{E}[\epsilon_{r(k)}(k) \mid F] = 0$.

Regarding communication error, the following assumptions define the restrictions on perturbation vector $\mathbf{m}(k)$ and random matrix $\mathbf{B}(k)$.

Assumption 3 *With probability 1, we have:*

(a) $\mathbb{E}[\mathbf{m}(k) \mid F, s(k), r(k), \mathbf{B}(k)] = \mathbb{E}[\mathbf{m}(k)]$ *for all k.*

(b) $\mathbb{E}[\mathbf{m}(k)] = 0$ *and* $\mathbb{E}[\|\mathbf{m}(k)\|^2 \mid F] = \sigma_k^2$ *for all k and for some $\sigma_k \ge 0$.*

(c) $\mathbb{E}[\mathbf{B}(k) \mid F, s(k), r(k), \mathbf{m}(k)] = \mathbb{E}[\mathbf{B}(k)]$ *for all k.*

3. CONVERGENCE RESULTS

We must first examine the conditions of the communication noise that will allow the node values to converge towards a common value. This convergence of values depends on the characteristics of the perturbation element $\mathbf{B}(k)\mathbf{m}(k)$. It turns out that a sufficient condition for convergence is when $\sum_{k=1}^{\infty} \lambda_1\{\mathbb{E}[\mathbf{B}(k)^T\mathbf{B}(k)]\}\mathbb{E}[\|\mathbf{m}(k)\|^2] < \infty$ and this does not require any assumption on the convexity of functions f_i.

Theorem 1 *Suppose that*

$$\sum_{k=1}^{\infty} \lambda_1\{\mathbb{E}[\mathbf{B}(k)^T\mathbf{B}(k)]\}\mathbb{E}[\|\mathbf{m}(k)\|^2] < \infty. \quad (6)$$

Then with probability 1*, we have* $\sum_{k=1}^{\infty} \frac{\|\mathbf{x}(k)-y(k)\mathbf{1}\|}{k} < \infty$ *and* $\lim_{k\to\infty} \|\mathbf{x}(k) - y(k)\mathbf{1}\| = 0$*. In the case where* $\sigma_k = \sigma$ *for some* $\sigma > 0$*, our condition reduces to*

$$\sum_{k=1}^{\infty} \lambda_1\{\mathbb{E}[\mathbf{B}(k)^T\mathbf{B}(k)]\} < \infty.$$

With the above theorem, we can now state the convergence results for the case where the individual f_i's are convex. These functions do not need to be differentiable; where a gradient does not exist, a subgradient is used instead. The subgradient $\nabla g(x)$ of a function g at x is a vector that satisfies

$$\nabla g(x)^T(y - x) \leq g(y) - g(x), \quad (7)$$

for all y in the domain of g. The following theorem shows that the iterates of the algorithm converge to the solution of (1).

Theorem 2 *Let* $X^* = Argmin_{x\in R} f(x)$ *be non-empty, and* $f_i(x)$ *be convex for each* $i \in V$*. If*

$$\sum_{k=1}^{\infty} \lambda_1\{\mathbb{E}[\mathbf{B}(k)^T\mathbf{B}(k)]\}\mathbb{E}[\|\mathbf{m}(k)\|^2] < \infty, \quad (8)$$

Then, with probability 1*, the sequences* $\{x_i, k\}$*,* $i \in V$*, converge to the same point in* X^**.*

4. NUMERICAL EXAMPLE

With the result of convergence to a minimum for convex functions, we can now apply the gossip subgradient optimization algorithm to a practical example. In this section, we take a look at the problem of vector estimation. The problem of vector estimation can be given by

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (9)$$

where $\mathbf{x} \in \mathbb{R}^N$ is the vector that we want to estimate or detect, $\mathbf{A} \in \mathbb{R}^{M\times N}$ is the detection or sensing matrix and $\mathbf{y} \in \mathbb{R}^M$ is the data vector. If $M >$ rank$(\mathbf{A})$, then $\mathbf{x}$ can be deduced from both $\mathbf{y}$ and $\mathbf{A}$ by basic linear algebra, so detection is necessary only when $M <$ rank$(\mathbf{A})$.

One method of estimating the vector $\mathbf{x}$ is the least-absolute shrinkage and selection operator [10], also known as the Lasso. This technique was first used for estimation and continuous variable selection purposes in linear regression problems. The Lasso is a convex optimization problem described as follows:

$$\arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1. \quad (10)$$

Here, the ℓ_1 norm is the sum of the absolute values of the entries of x, and parameter $\mu \geq 0$ is a constant that controls the amount of shrinkage over the solution of (10) that is effected by the ℓ_1 norm sparsity-encouraging penalty. Solving the optimization problem (10) gives us an estimate for $\mathbf{x}$ based on the knowledge of $\mathbf{y}$ and $\mathbf{A}$.

Suppose now that there are J nodes, indexed 1 to j, and each node has a detection matrix $\mathbf{A}_j \in \mathbb{R}^{M_j\times N}$ that it uses to estimate $\mathbf{x}$, yielding data vector $\mathbf{y}_j \in \mathbb{R}^{M_j}$. Individually, these nodes can reproduce their own estimate of $\mathbf{x}$. However, they can also cooperate to detect $\mathbf{x}$ so that the resultant estimate is more accurate. One such method of cooperation is known as the distributed Lasso [11], or the D-Lasso. Let $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_J)$ and $\mathbf{A} = (\mathbf{A}_1^T, \ldots, \mathbf{A}_J^T)^T$. If there exists a central fusion center that can collect data from all the nodes, we can basically collect all the data vectors $\mathbf{y}_j$ and detection matrices $\mathbf{A}_j$ and then solve for (10), since this allows us to utilize all the information available in each of the nodes. Without a fusion center, gathering information from every node can be difficult, especially when the network is large. In this case, we can approach the problem in a consensus manner. Let $f_j(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}_j - \mathbf{A}_j\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1$. Thus, (10) can be rewritten as

$$\begin{aligned} &\text{minimize} \quad f(x) := \sum_{j=1}^{J} f_j(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in \mathbb{R}^N. \end{aligned} \quad (11)$$

This is the exact form that is found in (1), with each of the functions f_j being convex. The main difference is that $\mathbf{x}$ here is N-dimensional instead of 1-dimensional. However, because the algorithm can be applied elementwise, our results still hold for N-dimensional $\mathbf{x}$. Thus, our distributed stochastic optimization algorithm can applied to the Lasso problem in noisy networks.

In the following, we use a model of four nodes with an underlying complete graph topology. The four nodes work together to detect the value of a vector $\mathbf{x}$. We set $\mathbf{x}$ to be a sparse vector with dimension 20 and sparsity 2, that is, it has two random nonzero entries. We let the perturbation vector $\mathbf{m}(k)$ to be a Gaussian random vector and the matrix $\mathbf{B}(k)$ be the diagonal matrix with entries $\left(\frac{1}{2}\right)^k$. In this way, the perturbation vector and matrix satisfies the conditions of our theorems, which is $\sum_{k=1}^{\infty} \lambda_1\{\mathbb{E}[\mathbf{B}(k)^T\mathbf{B}(k)]\}\mathbb{E}[\|\mathbf{m}(k)\|^2] < \infty$. We set the initial iterates to be zero. We run one simulation with the nodes experiencing communication noise whilst detecting $\mathbf{x}$, and a second one without communication noise. At each time step, a node is chosen at random, and this node chooses one of the other three nodes with uniform probability to communicate with. To ensure greater comparison reliability, the two nodes that are communicated with each other at each time step for both simulations are ensured to be the same. Thus, $s(k)$ and $r(k)$ are the same for both sets of simulation. We then plot the mean squared error of each iteration.

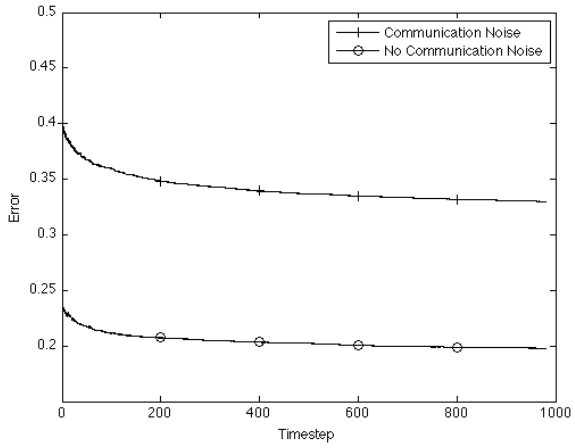

Fig. 1. Plot of the mean squared error with and without communication noise.

From Fig. 1, we see that the mean squared error decays as the iteration increases for both cases with and without noise. However, the presence of communication noise leads to a much higher mean squared error. Moreover, the convergence towards to the actual date vector is faster for the case without noise. However, this rate of convergence depends on how the perturbation matrix and vector are chosen. Note that both cases eventually converge towards $\mathbf{x}$ as the number of iterations increases, as proven by our theorems. Therefore, even if the individual nodes are not able to perform the optimization, a knowledge of the individual subgradient function is sufficient for the nodes to cooperate and iteratively arrive at a reasonable solution depending on requirement of the problem.

5. CONCLUSION

In this paper, we have presented results that show that the subgradient optimization algorithm does cause node values to converge under certain types of noisy communication environments. We also showed that for convex functions, the algorithm converges towards the solution of the optimization problem. We showed numerically that the algorithm indeed converges for the Lasso problem. Future work can be geared towards investigating the convergence properties for other kinds of noisy conditions and also other types of functions applicable in real-life scenarios.

6. REFERENCES

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[2] D.W. Soh, T.Q.S. Quek, and W.P. Tay, "Randomized broadcast in dynamic network environments," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Commun.*, San Francisco, CA, Jun. 2011, pp. 526–530.

[3] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[4] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[5] S.S. Ram, A. Nedić, and V.V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proc. of the IEEE Conference on Decision and Control*, Shanghai, Dec. 2009, pp. 3581–3586.

[6] S.S. Ram, A. Nedić, and V.V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[7] B. Johansson, *On distributed optimization in networked systems*, Ph.D. thesis, Royal Institute of Technology, 2008.

[8] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1291–1306, Jun. 2008.

[9] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, "Fully distributed algorithms for convex optimization problems," in *Proc. 21st Int. Symp. on Distrib. Comput.*, Berlin, Heidelberg, Sep. 2007, pp. 492–493.

[10] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Royal Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[11] J.A. Bazerque, G. Mateos, and G.B. Giannakis, "Distributed Lasso for in-network linear regression," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Dallas, Tx, Jun. 2010, pp. 2978–2981.