Distributed Compressed Sensing in Dynamic Networks

Stacy Patterson
Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA
sep@cs.rpi.edu

Yonina C. Eldar, Idit Keidar
Department of Electrical Engineering
Technion, Haifa, Israel
{yonina,idish}@ee.technion.ac.il

***Abstract*—We consider the problem of in-network compressed sensing, where the goal is to recover a global, sparse signal from local measurements using only local computation and communication. Our approach to this distributed compressed sensing problem is based on the centralized Iterative Hard Thresholding algorithm (IHT). In time-varying networks, the network dynamics necessarily introduce inaccuracies that are not present in a centralized implementation of IHT. To accommodate these inaccuracies, we show how centralized IHT can be extended to include inexact computations while still providing the same recovery guarantees. We then leverage these new theoretical results to develop a distributed version of IHT for dynamic networks. Evaluations show that our algorithm outperforms the best-known existing solution in both time and bandwidth by several orders of magnitude.**

***Index Terms*—distributed algorithm, iterative hard thresholding, distributed consensus**

I. Introduction

Recently there has been a great of deal of interest of interest in compressed sensing over networks, including applications such as event detection in sensor networks [1], traffic monitoring in vehicle networks [2], and collaborative cognitive radio networks [3]. In these settings, linear measurements of a sparse signal are taken by participants or *agents* in a network, and the goal is for all agents to recover the signal from their collective measurements. Since the measurements are distributed throughout the network, it is desirable to perform this recovery within the network in a distributed fashion.

Several recent works have proposed algorithms for distributed compressed sensing in static networks [3]–[6]. In sensor networks and radio networks, messages can be lost due to interference, and in vehicle networks, the network topology changes due to mobility. Thus, it is important for distributed compressed sensing algorithms to accommodate networks with time-varying topologies.

As far as we are aware, the only previously proposed algorithm that can be applied to distributed compressed sensing in a time-varying network is the distributed subgradient algorithm [7]. While this algorithm converges to the optimal solution of the compressed sensing problem, its convergence rate has been observed to be slow, and thus it incurs a high bandwidth cost. This cost is problematic in resource-constrained networks like sensor and radio networks.

We propose an alternative approach to distributed compressed sensing that is based on *Iterative Hard Thresholding* (IHT) [8]. In a centralized setting, IHT offers the benefit of computational simplicity when compared to many other recovery methods. Our distributed approach maintains this same computational benefit. In addition, recent work [9] has established that centralized IHT can be used for problems beyond compressed sensing, for example sparse signal recovery from nonlinear measurements. Our distributed solution provides the same recovery guarantees as centralized IHT and thus can also be applied to these settings.

In our distributed implementation of IHT, each agent stores an estimate of the signal In each iteration, every agent first performs a *simple local computation* to derive an intermediate vector. The agents then perform a *global computation* on their intermediate vectors to derive the next iterate. For networks that are time-varying, it is not possible to perform the global computation exactly, however, it is possible to approximate the global computation using only local communication.

We first show how centralized IHT can be extended to accommodate inexact computations while still providing the same recovery guarantees as the original IHT formulation. We then leverage these new theoretical results to develop a distributed IHT algorithm that uses multiple rounds of a distributed consensus algorithm to execute each inexact global computation. We call this algorithm *consensus-based distributed IHT* (CB-DIHT). Our approach was inspired by recent work on a distributed proximal gradient technique [10] that also uses multiple rounds of distributed consensus to perform an inexact computation in each iteration. However, this algorithm depends on assumptions that are not satisfied by the compressed sensing problem. We evaluate the performance of CB-DIHT on several example problems and show that it requires several orders of magnitude less time and bandwidth for recovery than the distributed subgradient algorithm.

The remainder of this work is organized as follows. In Section II, we present our system model and problem formulation. In Section III, we present our distributed algorithm. Finally, Section IV gives our simulation results.

II. Problem Formulation

We consider a network of P agents. The agents may be sensors or they may be fusion nodes that collect measurements from several nearby sensors. We assume there is a unique agent identified as agent 1. If this agent is not defined a priori, one can be chosen using a variety of well-known distributed algorithms (see [11]).

At each time step t, the network is modeled by a directed graph $(V, E^{(t)})$, where V is the set of P agents and $E^{(t)}$ is the set of directed communication links between them at time t. Messaging is reliable and synchronous, meaning that any message sent in time t is received before time $t+1$. We adopt the following standard assumption about the network connectivity over time [10], [12], [13].

Assumption 1: The sequence of graphs $\{(V, E^{(t)})\}_{t\geq 0}$ satisfies the following conditions:

1) The graph $(V, E^{(\infty)})$ is strongly connected, where $E^{(\infty)}$ is the set of edges that appear in infinitely many time steps.
2) There exists an integer $C \geq 1$ such that, if $(q,p) \in E^{(\infty)}$, then $(q,p) \in E^{(t)} \cup E^{(t+1)} \cup \cdots \cup E^{(t+C-1)}$, for all $t \geq 0$.

In short, this assumption means that, while the network may not be connected in any given time step, the union of graphs over each interval of C consecutive time steps is a strongly connected graph. The agents do not know C.

The agents seek to estimate a signal $x \in \mathbb{R}^N$ that is K-sparse, meaning x has at most K non-zero elements. Each agent has one or more (possibly noisy) measurements of the signal, and each has a loss function $f_p : \mathbb{R}^N \to \mathbb{R}$, known only to agent p, that indicates how well a given vector satisfies its measurements. As a specific example, we consider compressed sensing with linear measurements [14]. A more general problem formulation is considered in the technical report [18]. In compressed sensing with linear measurements, each agent p has M_p linear measurements of x taken using its sensing matrix $A_p \in \mathbb{R}^{M_p \times N}$. The total number of measurements is $M = \sum_{p=1}^{P} M_p$. The measurement vector of agent p, denoted b_p, is given by $b_p = A_p x + e_p$, where $e_p \in \mathbb{R}^{M_p}$ is the measurement error. The loss function for agent p is $f_p(x) := \|A_p x - b_p\|_2^2$, and the global loss function is the sum of these individual loss functions,

$$f(x) := \sum_{p=1}^{P} \|A_p x - b_p\|_2^2 = \|Ax - b\|_2^2,$$

where $A := \left[A_1^{\mathsf{T}} | \cdots | A_P^{\mathsf{T}}\right]^{\mathsf{T}}$ and $b := \left[b_1^{\mathsf{T}} | \cdots | b_P^{\mathsf{T}}\right]^{\mathsf{T}}$.

The goal is for every agent to recover the same signal x from their collective measurements using only communication between neighbors in the network. To recover x, the agents attempt to solve the optimization problem,

$$\text{minimize } f(x) := \sum_{p=1}^{P} f_p(x) \quad \text{subject to} \quad \|x\|_0 \leq K, \tag{1}$$

where $\|\cdot\|_0$ denotes the ℓ_0 norm, i.e., the number of non-zero components. Each agent only knows its own measurements, and so the agents must collaborate to solve this problem.

III. Algorithm

Problem (1) is NP-Hard in general [15]. However, for suitable loss functions, efficient centralized algorithms exist. Our distributed recovery algorithm is based on one of these, Iterative Hard Thresholding (IHT) [8], [16], which is defined as follows. Let $\mathcal{T}_K(v)$ be the thresholding operator which returns a vector where all but the K entries of v with the largest magnitude are set to 0 (with ties broken arbitrarily). IHT begins with an arbitrary K-sparse vector $x^{(0)}$. In each iteration, a gradient-step is performed, followed by application of the thresholding operator. This iteration is given by,

$$x^{(k+1)} = \mathcal{T}_K\left(x^{(k)} - \tfrac{1}{L}\nabla f(x^{(k)})\right), \tag{2}$$

with $L > 2\lambda_{max}(A^{\mathsf{T}}A)$. Here, $\lambda_{max}(\,\cdot\,)$ denotes the largest eigenvalue of the matrix.

In a recent work [6], we presented a distributed implementation of IHT (DIHT) for static networks. Every agent stores an identical copy of the signal estimate $x^{(k)}$. In iteration k, each agent first performs a local computation to derive an intermediate vector,

$$z_p^{(k)} = \nabla f_p(x^{(k)}). \tag{3}$$

The agents then perform a global computation on their intermediate vectors to derive the next iterate,

$$x^{(k+1)} = \mathcal{T}_K\left(x^{(k)} - \tfrac{1}{L}\textstyle\sum_{p=1}^{P} z_p^{(k)}\right), \tag{4}$$

which is, again, identical at every agent. These local and global computations are identical to the IHT iteration in (2).

The global computation (4) requires the agents to compute the sum of their intermediate vectors. In a time-varying network, it is not possible for the agents compute this sum exactly in finite time using *any* algorithm without a priori knowledge of the network dynamics. This is because, without this knowledge, an agent cannot determine when it has received the information it needs (from all other agents) to compute the sum. We extend DIHT to time-varying networks by employing a distributed consensus algorithm [17] to *approximate* the average of the intermediate vectors and then using this approximation in the global computation.

To use distributed consensus in DIHT, we must first consider the effects of such approximation errors on the correctness of the centralized IHT algorithm. We capture these approximations in the form of inexact computations of the gradient ∇f. We next show that, under a limited assumption on the accuracy of the gradient values, IHT with inexact gradients provides the same recovery guarantees as IHT with exact gradient computations. We then show how we leverage these new theoretical results to develop a consensus-based DIHT algorithm for time-varying networks.

Centralized IHT with Inexact Gradients. The algorithm is initialized with a K-sparse vector $x^{(0)}$ for which $f(x^{(0)})$ is finite. In each iteration, an approximate gradient step is computed, followed by the application of the thresholding operator. The iteration is thus given by,

$$x^{(k+1)} = \mathcal{T}_K\left(x^{(k)} - \tfrac{1}{L}\left(\nabla f(x^{(k)}) + \epsilon^{(k)}\right)\right), \tag{5}$$

Here, $\epsilon^{(k)} \in \mathbb{R}^N$ is the error in the gradient computation in iteration k.

So long as the sequence $\{\|\epsilon^{(k)}\|^2\}_{t\geq 0}$ is summable, algorithm (5) provides the same recovery guarantees as IHT with exact gradients. Namely, (5) converges to an L-stationary point of problem (1), which is defined as follows.

Definition 1: For a given $L > 0$, a K-sparse vector $x^* \in \mathbb{R}^N$ is an *L-stationary point* of problem (1) if,

$$x^* = \mathcal{T}_{K}\left(x^* - \tfrac{1}{L}\nabla f(x^*)\right).$$

It has been shown that L-stationarity is a necessary condition for optimality (see [9], Theorem 3.3).

Theorem 3.1: Let A satisfy the K-regularity property, i.e., for every index set $I \subseteq \{1, 2, \ldots, N\}$ with $|I| = K$, the columns of A associated with the index set I are linearly independent. Let $\{x^{(k)}\}_{k\geq 0}$ be the sequence generated by (5) with $L > 2\lambda_{max}(A^{\mathsf{T}}A)$ and with a sequence $\{\epsilon^{(k)}\}_{t\geq 0}$ satisfying $\sum_{k=0}^{\infty}\|\epsilon^{(k)}\|^2 < \infty$. Then, $\{x^{(k)}\}_{k\geq 0}$ converges to an L-stationary point of problem (1).
The proof of this theorem is given in the technical report [18].

Consensus-Based DIHT. We now detail our CB-DIHT algorithm. Each agent has an identical estimate $x_p^{(0)} = x_{init}$ for which $f_p(x_{init}) < \infty$, $p = 1 \ldots P$. The agents maintain identical estimates as the algorithm progresses. For each iteration k, the agents compute their intermediate vectors according to (3). They then execute a distributed consensus algorithm to compute an approximation of the average of these vectors.

Multiple iterations of the consensus algorithm are executed in each iteration of CB-DIHT, and one iteration of consensus corresponds to a single time step t in the time-varying network model. In the consensus algorithm instance for iteration k of CB-DIHT, every agent has a vector-valued state, initialized to its intermediate vector $v_p^{(0)} = z_p^{(k)}$. In each time step t, every agent computes a weighted average of its value and that of its neighbors in that time step. The vector at agent p evolves as,

$$v_p^{(t+1)} = \sum_{q=1}^{P} w_{pq}^{(t)}\ v_q^{(t)}, \tag{6}$$

where $w_{pq}^{(t)}$ is the weight that agent p assigns to the value at agent q. Under appropriate assumptions about the weights and the network connectivity over time (e.g. Assumption 1), the agents' vectors converge geometrically to the average of the initial vectors [12], [13].

After agent 1 executes $s^{(k)} = \lceil (k + \|x_1^{(k)}\|^2)/2 \rceil$ time steps of the consensus algorithm, it uses its local estimate of the average, denoted $\hat{v}^{(t)}$, to generate the next iterate $x_1^{(k+1)}$ as,

$$x_1^{(k+1)} = \mathcal{T}_{K}\left(x_1^{(k)} - \tfrac{1}{L_{TV}}\hat{v}^{(k)}\right), \tag{7}$$

with $L_{TV} > \frac{2}{P}\lambda_{max}(A^{\mathsf{T}}A)$. The value $x_1^{(k+1)}$ is then distributed to all other agents using a simple broadcast algorithm. Here, an agent, once it has received the value, sends this value once along all of its outgoing edges, excepting those edges on which it has previously received the value. On receipt of this value, an agent updates its local estimate $x_p^{(k+1)}$ to be this value, completing one iteration of CB-DIHT.

TABLE I: Recovery problem parameters.

Problem	N	M	P	K	$\lambda_{max}(A^{\mathsf{T}}A)$	L_{TV}
Sparco 902	1000	200	50	3	1	2.01/50
Sparco 7	2560	600	40	20	1	2.01/50
Sparco 11	1024	256	64	32	≈ 2283	4570/64, 600/64

In a time-varying network, it is not possible for the agents to independently determine when the value $x_1^{(k+1)}$ has been received by all other nodes, and therefore, when to start the next instance of distributed consensus. To overcome this challenge, we have developed a modified consensus algorithm that we call *diffusive distributed consensus* that combines the broadcast of an iterate $x_1^{(k)}$ with the consensus algorithm for iteration k. Rather than all agents beginning the consensus algorithm at the same time, agents begin participating the consensus algorithm once they have received the iterate. We show that diffusive distributed consensus also converges geometrically to the average of the intermediate vectors. For details of diffusive distributed consensus and CB-DIHT, we refer the reader to the technical report [18].

Algorithm Analysis. For each iteration k, the estimates at all agents $p \neq 1$ are identical to those at agent 1. It is straightforward to show that the evolution of $x_1^{(k)}$ can be formulated as an execution of centralized IHT with inexact gradients. The error in each iteration $\epsilon^{(k)}$ is the difference between agent 1's approximation of the average and the true average of the intermediate vectors. Under Assumption 1, the sequence of errors is square summable. Thus, CB-DIHT converges to an L-stationary point of problem (1). This result is formalized in the following theorem, a proof of which can be found in the technical report [18].

Theorem 3.2: Let Assumption 1 hold and let A satisfy the K-regularity property. Then the sequences $\{x_p^{(k)}\}_{k\geq 0}$, $p = 1 \ldots P$, generated by CB-DIHT with $L_{TV} > \frac{2}{P}\lambda_{max}(A^{\mathsf{T}}A)$ converge an L-stationary point of problem (1). Furthermore, all sequences converge to the same L-stationary point.

IV. Simulation Results

In this section, we present an experimental comparison of CB-DIHT and the distributed subgradient algorithm. We show evaluation results for three compressed sensing problems from the Sparco toolbox [19]. Details of the problems are given in Table I. For each problem, we divide the measurements evenly so that each agent has M/P measurements.

We evaluate each algorithm's performance on five different classes of graphs. For each class, we generate five random instances. The results shown in this section are the averages of the five runs over the five instances. The first graph type is a Barabasi-Albert (BA) scale free graph. The second and third graphs are Erdös-Rényi (ER) random graphs where each pair of vertices is connected with probability $pr = 0.25$ and probability $pr = 0.75$, respectively. The fourth and fifth graphs are geometric graphs with vertices placed uniformly at random in a unit square. In the fourth graph, two vertices are connected if they are within a distance of $d = 0.5$ of each other, and in

TABLE II: Number of distributed consensus iterations needed by CB-DIHT and the subgradient algorithm for signal recovery to accuracy of 10^{-2}.

	Sparco 902		Sparco 7		Sparco 11		
Graph	**CB-DIHT**	**Subgradient**	**CB-DIHT**	**Subgradient**	**CB-DIHT** ($L_{TV} = 4750/64$)	**CB-DIHT** ($L_{TV} = 600/64$)	**Subgradient**
BA	2.2×10^3	$> 3 \times 10^5$	3.0×10^3	$> 3 \times 10^5$	6.7×10^4	2.3×10^3	$> 3 \times 10^5$
ER ($pr = 0.25$)	2.5×10^3	$\geq 2.0 \times 10^5$	2.0×10^4	$> 3 \times 10^5$	7.1×10^4	8.1×10^2	$> 3 \times 10^5$
ER ($pr = 0.75$)	1.8×10^3	7.4×10^3	3.3×10^3	4.0×10^4	5.8×10^4	7.7×10^2	$> 3 \times 10^5$
Geo ($d = 0.5$)	6.5×10^3	$> 3 \times 10^5$	4.9×10^4	$> 3 \times 10^5$	1.0×10^5	5.7×10^3	$> 3 \times 10^5$
Geo ($d = 0.75$)	1.8×10^3	1.3×10^4	3.5×10^3	$\geq 6.8 \times 10^4$	5.2×10^4	8.7×10^2	$> 3 \times 10^5$

the fifth, vertices are connected if they are within a distance of $d = 0.75$. To make a time-varying network, for each graph, we choose ten random subgraphs, ensuring that the union of these subgraphs is the original graph. We cycle through these ten subgraphs, one per time step.

We have implemented all algorithms in Matlab. For CB-DIHT, we use the values of L_{TV} in Table I. For Sparco problem 11, $L_{TV} = 64/600$ is not sufficient to guarantee convergence, however, our evaluations show that, in all cases, CB-DIHT converged to the optimal solution. For the distributed subgradient algorithm, we experimented with different step-sizes $\eta^{(k)} = \frac{1}{k^a}$, where $a \in \{0.51, 0.6, 0.7, 0.8, 0.9, 1\}$. No single value of a performed best in all graphs. We therefore use $a = 0.7$, which is the value with the second fastest convergence rate for the vast majority of graphs.

Both CB-DIHT and the subgradient algorithm use distributed consensus as a building block; in the subgradient algorithm, agents execute one consensus round per iteration. In CB-DIHT, multiple diffusive consensus rounds are performed for each iteration. We compare the algorithms by counting the number of consensus rounds needed for $\|x_p^{(t)} - x^*\|/\|x^*\|$ to be less than 10^{-2} at every agent. For CB-DIHT, x^* may be an L-stationary point that is not an optimal solution, and we indicate when this occurs below. The subgradient algorithm solves a convex relaxation of problem (1) and will converge to the optimal solution.

The results are shown in Table II. We ran each experiment for a maximum of 3×10^5 consensus rounds. For problems 902 and 7, CB-DIHT converged to the optimal solution in every instance. In problem 902, CB-DIHT outperformed the subgradient algorithm by as much as two orders of magnitude. For Sparco problem 7, CB-DIHT required at least one order of magnitude fewer consensus rounds.

For problem 11 with $L_{TV} = 4750/64$, CB-DIHT converged to a suboptimal L-stationary point in the majority of experiments. For $L_{TV} = 600/64$, CB-DIHT always converged to the optimal solution. For both values of L_{TV}, CB-DIHT required fewer consensus rounds than the subgradient algorithm. This difference is more pronounced with $L_{TV} = 600/64$, where CB-DIHT outperformed the subgradient algorithm by at least two orders of magnitude. These results indicate a need for further investigation into the relationship between L_{TV} and the convergence of CB-DIHT.

Acknowledgement

This work is funded in part by: a Technion fellowship, the Lady Davis Fellowship Trust, the Israel Science Foundation under grant no. 1549/_11, the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI), and the Technion Autonomous Systems Program.

References

[1] J. Meng, H. Li, and Z. Han, "Sparse event detection in wireless sensor networks using compressive sensing," in *Proc 43rd Ann. Conf. Information Sciences and Systems*, 2009.

[2] Z. Li, Y. Zhu, H. Zhu, and M. Li, "Compressive sensing approach to urban traffic sensing," in *Proc. 31st Int. Conf. Distributed Computing Systems*, 2011, pp. 889–898.

[3] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Sig. Proc.*, vol. 58, no. 3, pp. 1847–1862, 2010.

[4] J. Mota, J. Xavier, P. Aguiar, and M. Püschel, "Basis pursuit in sensor networks," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Proc.*, 2011, pp. 2916–2919.

[5] ——, "Distributed basis pursuit," *IEEE Trans. Sig. Proc.*, vol. 60, no. 4, pp. 1942–1956, Apr 2012.

[6] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed sparse signal recovery in sensor networks," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Proc.*, 2013.

[7] I. Lobel, A. E. Ozdaglar, and D. Feijer, "Distributed multi-agent optimization with state-dependent communication," *Math. Program.*, vol. 129, no. 2, pp. 255–284, 2011.

[8] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[9] A. Beck and Y. C. Eldar, "Sparsity constrained nonlinear optimization: Optimality conditions and algorithms," *CoRR (to appear in SIAM Optimization)*, vol. abs/1203.4580, 2012.

[10] A. I. Chen and A. E. Ozdaglar, "A fast distributed proximal-gradient method," in *Proc. of Allerton Conference on Communication, Control, and Computing*, 2012.

[11] N. Lynch, *Distributed Algorithms*. USA: Morgan Kaufmann Publishers, Inc., 1996.

[12] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 2996–3000.

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

[14] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[15] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying newtorks," *CoRR*, vol. arXiv:1308.6086, 2013.

[16] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr 1995.

[17] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, Dec 2008.

[18] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. thesis, Massachusetts Institute of Technology, 1984.

[19] E. v. Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and Ö. Yılmaz, "Sparco: A testing framework for sparse reconstruction," Dept. Computer Science, University of British Columbia, Vancouver, Tech. Rep. TR-2007-20, October 2007.