Distributed Greedy Sparse Learning over Doubly Stochastic Networks

Ahmed Zaki, Arun Venkitaraman, Saikat Chatterjee and Lars K. Rasmussen
ACCESS Linneaus Center, KTH Royal Institute of Technology, Stockholm
Emails: zakiah@kth.se, arunv@kth.se, sach@kth.se, lkra@kth.se

***Abstract*—In this paper, we develop a greedy algorithm for sparse learning over a doubly stochastic network. In the proposed algorithm, nodes of the network perform sparse learning by exchanging their individual intermediate variables. The algorithm is iterative in nature. We provide a restricted isometry property (RIP)-based theoretical guarantee both on the performance of the algorithm and the number of iterations required for convergence. Using simulations, we show that the proposed algorithm provides good performance.**

I. Introduction

Two pillar members of signal processing, modeling and optimization for big data analysis are sparse learning and distributed algorithms [1]. For big data analysis, another aspect is the need of fast execution. In this context, low complexity distributed algorithms for large-scale sparse learning has a high potential [2], [3]. To realize large-scale distributed sparse learning, recent activities are reported in [4], [5] where greedy algorithms are designed and analyzed. A major advantage of greedy algorithms is their low computational complexity and hence their suitability for large-scale scenarios. However, the associated algorithmic steps are non-convex, and often has a limited analytical tractability.

In this article, we design a greedy algorithm for sparse learning that has low communication and computational costs, as well as a good theoretical foundation. The proposed algorithm exchanges estimates of underlying sparse signal over a doubly stochastic network, unlike the case of [4] where other data is also exchanged. The main contributions of this article are as follows:

1) We propose a greedy algorithm for distributed sparse learning, that is fast to converge, and has a low communication cost.
2) We carry out a restricted-isometry-property (RIP) based theoretical analysis. We also provide an upper bound on the number of times the nodes exchange information for convergence.

A. Relation to Prior Work

In this subsection, we provide a review of the relevant literature. A majority of the prior works have proposed convex optimization based algorithms for the problem of sparse learning over networks due to their theoretical tractibility. Distributed compressed sensing using convex optimization is addressed in [6], [7]. Using alternating-direction-method-of-multipliers (ADMM), distributed basis pursuit [8] and distributed LASSO (D-LASSO) [9] were realized. D-LASSO is shown to solve the exact convex optimization problem of a centralized scenario in a distributed fashion.

Our paper proposes a distributed greedy algorithm for sparse learning. In relation to that, a distributed iterative hard thresholding algorithm was proposed in [10] for distributed sparse learning in both static and time-varying networks. There, each node of a network finds a local intermediate estimate, and mean value of the set of estimates is found using a global consensus. Further improvement on the work in [10] was proposed in [11] to provide a reduced communication cost. Another distributed iterative hard thresholding (DiHaT) algorithm is developed in [4] where observations, measurement matrices and local estimates are exchanged over network to achieve consensus. The DiHaT algorithm provides fast convergence compared to D-LASSO, and provides competitive performance, but at the expense of a high communication cost. In [4], an alternate algorithm was also proposed that only uses estimate exchange, but without any theoretical analysis. Our algorithm can be implemented in more general networks with different measurement sizes at different nodes. In addition, the proposed algorithm has a faster convergence as it does not require consensus.

B. System Model, notations and preliminaries

We consider a connected network with L nodes. The neighborhood of node l is defined by the set $\mathcal{N}_l \subseteq \{1, 2, \ldots, L\}$. Each node is capable of receiving weighted data from other nodes in its neighborhood. The weights assigned to links between nodes can be written as a network matrix $\mathbf{H} = \{h_{i,j}\} \in \mathbb{R}^{L \times L}$, where $h_{i,j}$ is the link weight from j'th node to the i'th node. We assume that $\mathbf{H}$ is a doubly stochastic matrix. Our task is to learn or estimate an N-length sparse signal $\mathbf{x}$ in a distributed manner over the network. The observation vector $\mathbf{y}_l \in \mathbb{R}^{M_l}$ at the l'th node is modeled as

$$\mathbf{y}_l = \mathbf{A}_l \mathbf{x} + \mathbf{e}_l,$$

where $\mathbf{A}_l \in \mathbb{R}^{M_l \times N}$ is a measurement or dictionary matrix with $M_l < N$, and $\mathbf{e}_l \in \mathbb{R}^{M_l}$ is the additive noise. We assume that the sparse signal $\mathbf{x}$ has at most s non-zero scalar elements, and s is known a-priori. The above assumption is used for several greedy sparse learning algorithms, such as subspace pursuit (SP) [12] and CoSaMP [13].

We use the calligraphic letter $\mathcal{T}$ to denote sub-sets of $\Omega \triangleq \{1, 2, \ldots, N\}$. We use $|\mathcal{T}|$ and $\mathcal{T}^c$ to denote the

ISBN 978-0-9928626-7-1 © EURASIP 2017

cardinality and complement of the set $\mathcal{T}$, respectively. For the matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, a sub-matrix $\mathbf{A}_{\mathcal{T}} \in \mathbb{R}^{M\times|\mathcal{T}|}$ consists of the columns of $\mathbf{A}$ indexed by $i \in \mathcal{T}$. Similarly, for $\mathbf{x} \in \mathbb{R}^N$, a sub-vector $\mathbf{x}_{\mathcal{T}} \in \mathbb{R}^{|\mathcal{T}|}$ is composed of the components of $\mathbf{x}$ indexed by $i \in \mathcal{T}$. Also, we denote $(.)^t$ and $(.)^\dagger$ as transpose and pseudo-inverse, respectively. In this article $\mathbf{A}_{\mathcal{T}}^\dagger \triangleq (\mathbf{A}_{\mathcal{T}})^\dagger$. We use $\|.\|$ and $\|.\|_0$ to denote the standard ℓ_2 norm and ℓ_0 norm of a vector, respectively. For a sparse signal $\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots, x_N]^t$, the support-set $\mathcal{T}$ of $\mathbf{x}$ is defined as $\mathcal{T} = \{i : x_i \neq 0\}$. We define a function that finds the s-length support of a vector, as $\texttt{supp}(\mathbf{x}, s) \triangleq \{$the set of indices corresponding to the s largest amplitude components of $\mathbf{x}\}$. The s-Restricted Isometry Constant (RIC) [14] of a matrix $\mathbf{A}$ is denoted as δ_s.

The rest of the paper is organized as follows. The distributed greedy algorithm and the associated theoretical results are presented in Section II. Simulation results and discussions are shown in Section III. Finally, the conclusions are presented in Section IV.

II. Distributed Algorithm - NGP

In this section, we describe the proposed distributed greedy algorithm. We refer to the algorithm as network greedy pursuit (NGP). The algorithm is motivated by the algorithmic structures of SP and CoSaMP. The pseudo-code of the NGP algorithm is shown in Algorithm 1. The NGP algorithm

Algorithm 1 Network Greedy Pursuit - at l'th node

Known inputs: $\mathbf{y}_l$, $\mathbf{A}_l$, s, $\{h_{l,j}\}_{j=1}^L$
Initialization:

$k \leftarrow 0$ (k denotes time index)
$\mathbf{r}_{l,k} \leftarrow \mathbf{y}_l$ (Residual at k'th time)
$\hat{\mathcal{T}}_{l,k} \leftarrow \emptyset$ (Support-set at k'th time)
$\hat{\mathbf{x}}_{l,k} \leftarrow \mathbf{0}$ (Sparse solution at k'th time)

Iteration:

repeat
$k \leftarrow k + 1$ (Iteration counter)
1: $\grave{\mathcal{T}}_{l,k} \leftarrow \texttt{supp}(\mathbf{A}_l^t \mathbf{r}_{l,k}, s)$
2: $\tilde{\mathcal{T}}_{l,k} \leftarrow \grave{\mathcal{T}}_{l,k} \cup \hat{\mathcal{T}}_{l,k-1}$
3: $\tilde{\mathbf{x}}_{l,k}$ such that $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}} \leftarrow \mathbf{A}_{l,\tilde{\mathcal{T}}_{l,k}}^\dagger \mathbf{y}_l$; $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$
4: $\check{\mathbf{x}}_{l,k} \leftarrow \sum_{r\in\mathcal{N}_l} h_{l,r}\ \tilde{\mathbf{x}}_{r,k}$ (Information exchange)
5: $\hat{\mathcal{T}}_{l,k} \leftarrow \texttt{supp}(\check{\mathbf{x}}_{l,k}, s)$
6: $\hat{\mathbf{x}}_{l,k}$ such that $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}} \leftarrow \mathbf{A}_{l,\hat{\mathcal{T}}_{l,k}}^\dagger \mathbf{y}_l$; $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$
7: $\mathbf{r}_{l,k} \leftarrow \mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k}$
until *stopping criterion*

Final output: $\hat{\mathbf{x}}_l$, $\hat{\mathcal{T}}_l$, $\mathbf{r}_l$

is executed locally at each node of the network. Denoting k as the iteration counter for information exchange, the l'th node receives an intermediate estimate $\tilde{\mathbf{x}}_{r,k}, r \in \mathcal{N}_l$ from its neighboring nodes and performs the weighted sum (see step 4 of the algorithm). Note that $\tilde{\mathbf{x}}_{l,k}$ and $\check{\mathbf{x}}_{l,k}$ both can have a sparsity level higher than s. The support finding function, $\texttt{supp}(.)$ incorporates pruning such that the a-priori knowledge of the sparsity level is used to get an s-length estimate, $\hat{\mathbf{x}}_{l,k}$. The stopping criterion of the algorithm can be either a predetermined number of iterations or a non-decreasing ℓ_2-norm of the residual. The NGP algorithm is designed to improve performance via cooperation using information exchange compared to no cooperation. Now we state the main theoretical result of the algorithm.

Main result: In the NGP algorithm, the estimated sparse signal $\hat{\mathbf{x}}_{l,k}$ follows a recurrence inequality, stated in the following theorem.

Theorem 1: For the NGP algorithm, the estimate at the k'th iteration satisfies,

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| \leq \sum_{r\in\mathcal{N}_l} h_{l,r}\left(c_1\|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + c_2\|\mathbf{e}_r\|\right) + c_3\|\mathbf{e}_l\|,$$

and

$$\|\mathbf{x}_{\hat{\mathcal{T}}_{l,k}^c}\| \leq c_1 \sum_{r\in\mathcal{N}_l} \|\mathbf{x}_{\hat{\mathcal{T}}_{r,k-1}^c}\| + c_4 \sum_{r\in\mathcal{N}_l} \|\mathbf{e}_r\|,$$

where

$$c_1 = \sqrt{\tfrac{2\delta_{3s}^2(3-\delta_{3s}^2)}{(1-\delta_{3s}^2)^2}},$$
$$c_2 = \sqrt{\tfrac{2}{1-\delta_{3s}^2}}(c_5 + c_6), c_3 = \tfrac{\sqrt{1+\delta_{2s}}}{(1-\delta_{3s})},$$
$$c_4 = \sqrt{2}(c_5 + c_6) + \sqrt{\tfrac{2\delta_{3s}^2(3-\delta_{3s}^2)(1+\delta_{2s})}{(1-\delta_{3s}^2)^2(1-\delta_{3s})^2}} \text{ and}$$
$$c_5 = \tfrac{\sqrt{2(1-\delta_{3s})}+\sqrt{1+\delta_{3s}}}{1-\delta_{3s}}, c_6 = \sqrt{1+\delta_{3s}}.$$

The proof of the above theorem is not shown here for brevity and will be discussed in an extended manuscript later. In addition, we can compactly write the above recurrence inequality at all the nodes in the form shown below.

Corollary 1: For the NGP algorithm, the estimate at the k'th iteration satisfies,

$$\sum_{l=1}^L \|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| \leq c_1 \sum_{l=1}^L \|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + (c_2 + c_3) \sum_{l=1}^L \|\mathbf{e}_l\|,$$

and

$$\sum_{l=1}^L \|\mathbf{x}_{\hat{\mathcal{T}}_{l,k}^c}\| \leq c_1 \sum_{l=1}^L \|\mathbf{x}_{\hat{\mathcal{T}}_{l,k-1}^c}\| + c_4 \sum_{l=1}^L \|\mathbf{e}_l\|.$$

Proof The above result follows from Theorem 1 by summing up both sides $\forall l$ and using the doubly stochastic property of the network matrix, $\mathbf{H}$.

We next show that under some technical conditions and the condition $c_1 < 1$, the NGP algorithm will converge in a finite number of iterations. We have the following theorem on the convergence of NGP.

Theorem 2: If $\delta_{3s} < 0.362$ and $\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}} > 1$, then after $k^* = \lceil \log\left(\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}}\right) / \log\left(\frac{1}{c_1}\right) \rceil$, the performance of the NGP algorithm at the l'th node converges and is bounded by

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,k^*}\| \leq c_7 \|\mathbf{e}\|_{\max},$$

where $\|\mathbf{e}\|_{\max} = \max_l \|\mathbf{e}_l\|$ and $c_7 = \left(1 + \frac{c_2+c_3}{1-c_1}\right)$.

Proof From Theorem 1, at the k^*'th iteration, we can write in the vector form,

$$\underline{\|\mathbf{x} - \hat{\mathbf{x}}_{k^*}\|} \leq \quad c_1\mathbf{H}\underline{\|\mathbf{x} - \hat{\mathbf{x}}_{k^*-1}\|} + (c_3\mathbf{I}_L + c_2\mathbf{H})\,\underline{\|\mathbf{e}\|},$$

where $\underline{\|\mathbf{x}-\hat{\mathbf{x}}_k\|} \triangleq [\|\mathbf{x}-\hat{\mathbf{x}}_{1,k}\|\dots\|\mathbf{x}-\hat{\mathbf{x}}_{L,k}\|]^t$ and $\underline{\|\mathbf{e}\|} \triangleq [\|\mathbf{e}_1\|\dots\|\mathbf{e}_L\|]^t$. We assume component-wise inequality in the above expression. The above bound can be simplified as

$$\begin{aligned}\underline{\|\mathbf{x}-\hat{\mathbf{x}}_{k^*}\|} \leq\ & c_1\mathbf{H}\underline{\|\mathbf{x}-\hat{\mathbf{x}}_{k^*-1}\|} + (c_3\mathbf{I}_L + c_2\mathbf{H})\,\underline{\|\mathbf{e}\|}_{\max} \\ \overset{(a)}{\leq}\ & c_1\mathbf{H}\underline{\|\mathbf{x}-\hat{\mathbf{x}}_{k^*-1}\|} + (c_2+c_3)\underline{\|\mathbf{e}\|}_{\max},\end{aligned}$$

where (a) follows from the right stochastic property of $\mathbf{H}$ and $\underline{\|\mathbf{e}\|}_{\max} \triangleq [\|\mathbf{e}\|_{\max}\dots\|\mathbf{e}\|_{\max}]^t$. Applying the above relation iteratively and under the condition $c_1 < 1$, we get

$$\begin{aligned}\underline{\|\mathbf{x}-\hat{\mathbf{x}}_{k^*}\|} \leq\ & (c_1\mathbf{H})^{k^*}\,\underline{\|\mathbf{x}-\hat{\mathbf{x}}_0\|} \\ & +(c_2+c_3)\left(\mathbf{I}_L+\dots+(c_1\mathbf{H})^{k^*-1}\right)\underline{\|\mathbf{e}\|}_{\max} \\ \overset{(a)}{\leq}\ & c_1^{k^*}\,\underline{\|\mathbf{x}\|} + \tfrac{c_2+c_3}{1-c_1}\underline{\|\mathbf{e}\|}_{\max},\end{aligned}$$

where (a) follows from the right stochastic property of $\mathbf{H}$ and the initial condition, $\|\mathbf{x}-\hat{\mathbf{x}}_{l,0}\| = \|\mathbf{x}\|, \forall l$. Also, we have defined $\underline{\|\mathbf{x}\|} \triangleq [\mathbf{x}\dots\mathbf{x}]^t$ in the above equation. The technical condition $c_1 < 1$ translates to the requirement that δ_{3s} of $\mathbf{A}_l$ has to follow $\delta_{3s} < 0.362$. Substituting $k^* = \lceil \log\left(\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}}\right)/\log\left(\frac{1}{c_1}\right)\rceil$, we can write,

$$\underline{\|\mathbf{x}-\hat{\mathbf{x}}_{k^*}\|} \leq \left(1+\tfrac{c_2+c_3}{1-c_1}\right)\underline{\|\mathbf{e}\|}_{\max}.$$

The result follows from the component-wise inequality in the above equation.

Remark: As an example, for $\delta_{3s} < 0.20$ and SMNR (defined in Section III) of 20 dB, we have the number of iterations, $k^* = 7$ and $c_7 = 15.62$.

The above theorem signifies that if $\delta_{3s} < 0.362$ and there is no noise, then the NGP algorithm achieves exact estimate of $\mathbf{x}$ at every node.

III. Simulation and discussion

In this section, we study the performance of NGP algorithm. We first describe the simulation setup and the test scenarios.

A. Simulation Setup

For simulations, given a choice of an edge matrix of a network, $\mathbf{H}$ is computed via optimization of the *second largest eigenvalue modulus* (SLEM) of the matrix. This is known as the SLEM optimization problem in the literature [15]. For a fixed $\mathbf{H}$, we perform Monte-Carlo simulations where we randomly draw measurement matrices, sparse signals and additive Gaussian noise. We used mean signal-to-noise error (MSNR) as a performance metric, as defined below,

$$\text{MSNR} = \frac{1}{L}\sum_{l=1}^{L}\frac{\mathcal{E}\{\|\mathbf{x}\|^2\}}{\mathcal{E}\{\|\mathbf{x}-\hat{\mathbf{x}}_l\|^2\}}.$$

We consider that all measurement matrices across nodes are of the same size, that means $\forall l$, $M_l = M$. This condition is necessary for comparing with DiHaT [4]. Define signal-to-measurement-noise ratio (SMNR) for the l'th node as,

$$\text{SMNR}_l = \frac{\mathcal{E}\{\|\mathbf{x}\|^2\}}{\mathcal{E}\{\|\mathbf{e}_l\|^2\}}.$$

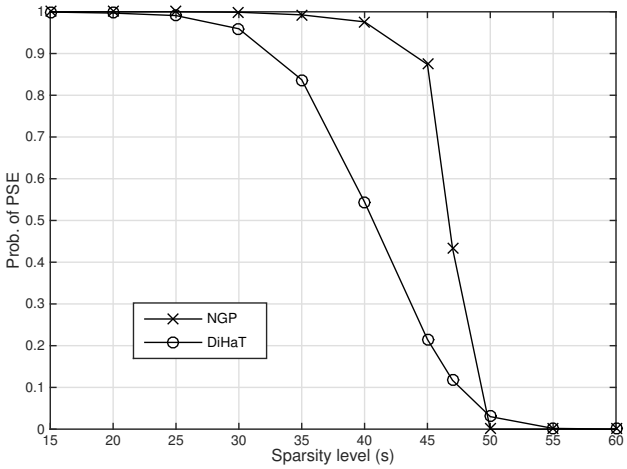

Fig. 1: Probability of PSE for NGP and DiHaT, binary sparse data, M = 100, N = 500, L = 20, d = 5, number of information exchanges = 30, no observation noise.

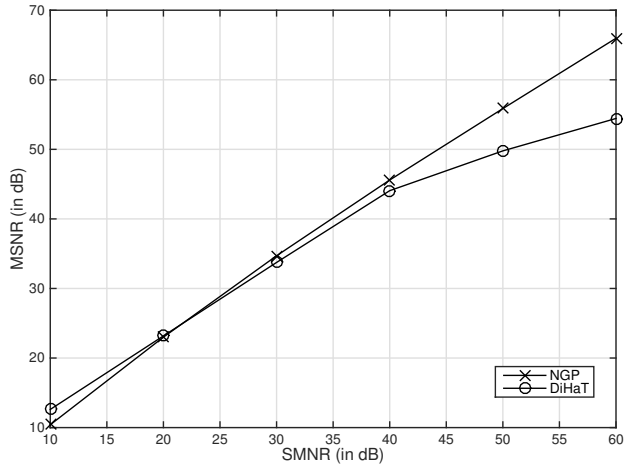

Fig. 2: MSNR of NGP and DiHaT with respect to SMNR, Gaussian sparse data, M = 100, N = 500, s = 20, L = 20, d = 5, number of information exchanges = 30.

We further assume that $\forall l$, SMNR_l are of the same value. In [4], the greedy DiHaT algorithm is compared with the convex D-LASSO and shown to provide faster convergence. In our simulations, we compare NGP with DiHaT. The stopping criterion of algorithms is a maximum allowable iterations that we set as 30. This number is motivated by the experiment in Section III-D.

We define a term called the degree of network to represent how densely the network is connected. Let us denote the degree of network by d. This means that each node of the network is connected with d other nodes or $\forall l$, $|\mathcal{N}_l| = d$.

ISBN 978-0-9928626-7-1 © EURASIP 2017

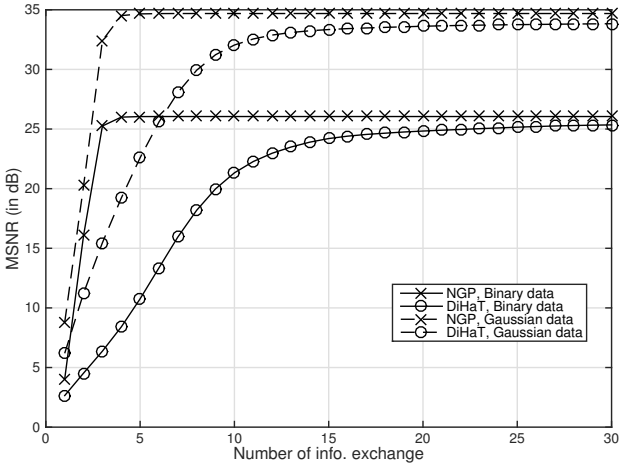

Fig. 3: MSNR of NGP and DiHaT with respect to the number of information exchanges, binary and Gaussian sparse data, M = 100, N = 500, s = 20, L = 20, d = 5, SMNR = 30 dB.

B. Phase transition experiment

In this experiment, we use binary sparse vector $\mathbf{x}$ such that the non-zero elements are set to ones. We are interested in the probability of perfect support-set estimation (PSE). Fixing $M = 100$ and $N = 500$, we vary s and compare between NGP and DiHaT. For this experiment, we used $L = 20$ nodes and each node is connected with five other nodes. The experiment requires noise-less observations. The PSE performance of the NGP and DiHaT is shown in Fig. 1. It is evident from the figure that the NGP can recover signals with a higher sparsity level as compared to the DiHaT. This can be explained partly by the higher RIC, $\delta_{3s} = 0.362$ for NGP as compared to $\delta_{3s} = 0.334$ for DiHaT [4].

C. Experiment for robustness to measurement noise

In this experiment, we check how performance of the two algorithms vary with respect to the SMNR. We used the same network matrix of $\mathbf{H}$ from section III-B, and $M = 100$, $N = 500$, $s = 20$. The signal is generated as a Gaussian sparse signal, where the non-zeros of $\mathbf{x}$ are i.i.d. Gaussian. We plot the MSNR with respect to the SMNR in Fig. 2. It can be seen that the performance of the NGP is better than that of the DiHaT.

D. Influence of information exchange

In this experiment, we compare the performance of NGP and DiHaT over iterations at 30 dB SMNR. We used the same network matrix of $\mathbf{H}$ as in section III-B. For this experiment, we used both binary and Gaussian sparse signals. Keeping $M = 100$, $N = 500$ and $s = 20$, we plot the MSNR performance of NGP and DiHaT with respect to the number of information exchanges in the network. The results are shown in Fig. 3. It can be seen that the NGP has a higher MSNR floor compared to DiHaT for both the cases of binary and Gaussian sparse signals. In addition, the NGP has a faster convergence (less than five iterations) as compared to the DiHaT (around 20 iterations). Note that the performance of both the NGP and DiHaT saturates after 30 iterations and hence the motivation for fixing the stopping criterion as 30 iterations.

IV. Conclusion

We developed a distributed greedy algorithm to solve the problem of sparse learning over networks. The proposed algorithm iteratively refines the estimate at a node with information from the neighboring nodes. The algorithm was shown to have an RIP-based convergence guarantee. Under simulations, the algorithm demonstrated a better estimation performance as compared to the state-of-the-art distributed greedy algorithms found in literature.

References

[1] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (statistical) learning tools for our era of data deluge," *IEEE Signal Processing Mag.*, vol. 31, no. 5, pp. 18–31, Sep. 2014.

[2] Z. J. Xiang, H. Xu, and P. J. Ramadge, "Learning sparse representations of high dimensional data on large scale dictionaries," in *Advances in Neural Information Processing Systems 24*. Curran Associates Inc., 2011, pp. 900–908. [Online]. Available: http://papers.nips.cc/paper/4400-learning-sparse-representations-of-high-dimensional-data-on-large-scale-dictionaries.pdf

[3] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt, "Recent developments in the sparse fourier transform: A compressed fourier transform for big data," *IEEE Signal Processing Mag.*, vol. 31, no. 5, pp. 91–100, Sep 2014.

[4] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "Greedy sparsity-promoting algorithms for distributed learning," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1419–1432, Mar. 2015.

[5] D. Sundman, S. Chatterjee, and M. Skoglund, "Design and analysis of a greedy pursuit for distributed compressed sensing," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2803–2818, Jun. 2016.

[6] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Conference Record of the Thirty-Ninth Asilomar Conf. on Signals, Systems and Computers*, Nov., 2005, pp. 1537 – 1541.

[7] D. Baron, M. Duarte, M. Wakin, S. Sarvotham, and R. Baraniuk, "Distributed compressive sensing," jan. 2009. [Online]. Available: http://arxiv.org/abs/0901.3403

[8] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[9] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[10] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4931–4946, Oct. 2014.

[11] P. Han, R. Niu, and Y. C. Eldar, "Modified distributed iterative hard thresholding," in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 3766–3770.

[12] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230 –2249, May 2009.

[13] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *California Inst. Tech., Tech. Rep.*, no. 2008-01, Jul. 2008.

[14] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203 – 4215, Dec. 2005.

[15] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, Apr. 2004. [Online]. Available: http://dx.doi.org/10.1137/S0036144503423264

ISBN 978-0-9928626-7-1 © EURASIP 2017