

Estimate Exchange over Network is Good for Distributed Hard Thresholding Pursuit

Ahmed Zaki*, Partha P. Mitra†, Lars K. Rasmussen* and Saikat Chatterjee*

* School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

† Cold Spring Harbor Laboratory, 1 Bungtown Road, New York, USA

Abstract—We investigate an existing distributed algorithm for learning sparse signals or data over networks. The algorithm is iterative and exchanges intermediate estimates of a sparse signal over a network. This learning strategy using exchange of intermediate estimates over the network requires a limited communication overhead for information transmission. Our objective in this article is to show that the strategy is good for learning in spite of limited communication. In pursuit of this objective, we first provide a restricted isometry property (RIP)-based theoretical analysis on convergence of the iterative algorithm. Then, using simulations, we show that the algorithm provides competitive performance in learning sparse signals vis-a-vis an existing alternate distributed algorithm. The alternate distributed algorithm exchanges more information including observations and system parameters.

I. INTRODUCTION

The topic of estimating and/or learning of sparse signals has many applications, such as machine learning [1], [2], multimedia processing [3], compressive sensing [4], wireless communications [5], just to mention a few. Here we consider a distributed sparse learning problem. Consider a network consisting of L nodes with a network matrix $\mathbf{H} \in \mathbb{R}^{L \times L}$ describing the connections among the nodes. The (l, r) 'th element h_{lr} of \mathbf{H} specifies the weight of the link from node r to node l . A zero valued h_{lr} signifies the absence of a direct link from node r to node l . In the literature, \mathbf{H} is also known as a network policy matrix. Let $\mathbf{x} \in \mathbb{R}^N$ be the underlying sparse signal to estimate/learn. Further assume that node l has the observation

$$\mathbf{y}_l = \mathbf{A}_l \mathbf{x} + \mathbf{e}_l. \quad (1)$$

Here \mathbf{e}_l is the error term and $\mathbf{A}_l \in \mathbb{R}^{M_l \times N}$ is the system matrix (a sensing matrix or dictionary, depending on a particular application) at node l . For distributed learning, the nodes of a network exchange various information, for example, intermediate estimates of \mathbf{x} , observations \mathbf{y}_l , system matrices \mathbf{A}_l , or their parameters. Using such information, a distributed algorithm learns \mathbf{x} at each node over iterations. Few important aspects of a distributed algorithm are scalability with the number of nodes, low computational requirements at each node, and limited communication between nodes. These aspects are required to realize a large distributed system with many nodes and high-dimensional system matrices. To comply with the low computational aspect, we focus on greedy algorithms. Standard greedy algorithms, such as orthogonal matching pursuit [6], subspace pursuit [7], CoSaMP [8], and

their variants [9], [10] are of low computational load and fast in execution. Further, for limited communication aspect, we prefer that intermediate estimates of \mathbf{x} are exchanged over the network. Therefore we focus on developing distributed greedy sparse learning algorithms that exchange intermediate estimates of \mathbf{x} or relevant parameters over network. Relevant past works in this direction are [11], [12], [13] where we proposed some rules for information exchange over network and developed new greedy algorithms for various signal models.

In this current article, we investigate an existing distributed greedy algorithm in the literature [14]. Our interest is to analyze performance of the algorithm. We do not propose any new rule or new algorithm. The existing algorithm of [14] exchanges intermediate estimates of \mathbf{x} between the nodes of the network. We refer to this algorithm as the distributed hard thresholding pursuit (DHTP). DHTP has not received due attention in [14] – minor simulation results were reported and no theoretical analysis was performed. For the DHTP, our main objective is to show that estimate-exchange is a good strategy for achieving a good learning performance. In pursuit of this objective, we provide theoretical analysis and extensive simulation results. Using simulations, we show that there is no significant incentive in performance gain due to exchange of observations \mathbf{y}_l and system matrices \mathbf{A}_l . Exchange of intermediate estimates over network is good and that helps in communication and computation constrained scenarios.

A. Literature survey

We provide a literature survey for the problem of sparse learning over network. For this problem, learning by consensus is a popular strategy. A consensus strategy achieves estimates that are all the same at network nodes after convergence. That means $\forall l, \hat{\mathbf{x}}_l = \hat{\mathbf{x}}$, where $\hat{\mathbf{x}}_l$ denotes the signal estimate at node l at convergence of a distributed sparse learning algorithm. Using a greedy learning approach, a consensus seeking algorithm was recently proposed in [14]. This algorithm is referred to as the distributed hard thresholding (DiHaT). To achieve consensus, the DiHaT exchanges intermediate estimates of \mathbf{x} , observations \mathbf{y}_l and system matrices \mathbf{A}_l . Furthermore, DiHaT uses a consensus seeking network matrix \mathbf{H} with special properties; DiHaT requires that \mathbf{H} be a doubly stochastic matrix. Here consensus seeking means that each and every node have estimates that are same at convergence. Alternatively, without seeking consensus, there exist several greedy sparse learning algorithms. DHTP of [14]

is one example. Other examples include algorithms from [15], [16], [12], [17], [18].

For distributed sparse learning, there exist several convex optimization based algorithms, mainly in the application area of distributed compressed sensing [19], [20]. Some of the algorithms provide a centralized solution using a distributed convex optimization algorithm called the alternating-direction-method-of-multipliers (ADMM) [21]. ADMM based distributed learning algorithms were proposed in [22], [23]. Specifically, the ADMM based method of [23] is called D-LASSO. It is worth mentioning that the D-LASSO is shown to provide a slower convergence compared to greedy DiHaT in [14]. Using adaptive signal processing techniques such as gradient search, distributed sparse learning and sparse regression were realized in [24], [25], [26]. These adaptive algorithms typically use a mean-square-error cost averaged over all the nodes in a network to find an optimal solution via gradient search. Distributed learning and regression are then performed via diffusion of information over a network and adaptation in all individual nodes. Using a Bayesian framework for finding the posterior with sparsity promoting priors, a distributed-message-passing based method was proposed in [27] and a sparse Bayesian learning based method was proposed in [28]. Further, to promote sparsity in solutions, distributed system learning such as distributed dictionary learning was also considered in [29], [30]. Next, we mention that there exist several signal models in the literature where sparse signals are not the same for all nodes of a network. For example, denoting the signal at node l by \mathbf{x}_l , supports of \mathbf{x}_l are the same in [17], [31], [32], [33], but not their signal values; further, \mathbf{x}_l have common and private support and/or signal parts in [34], [12], [18]. In this article, we consider the setup (1) where $\forall l, \mathbf{x}_l = \mathbf{x}$.

B. Contributions

Our objective is to show that signal estimate exchange is good for distributed sparse learning. There is no need to exchange \mathbf{y}_l and \mathbf{A}_l . In pursuit of this objective, we investigate DHTP and our contributions are as follows.

- 1) We provide a restricted-isometry-property (RIP) based theoretical analysis and convergence guarantee for DHTP. For error-free condition, that means $\forall l, \mathbf{e}_l = \mathbf{0}$, learned estimates at all nodes converge to the true signal \mathbf{x} under some technical conditions.
- 2) Using simulations, we show instances where DHTP provides better learning performance than DiHaT. For a fair comparison, we evaluate practical performance using doubly stochastic \mathbf{H} matrix.
- 3) We show that DHTP performs good for a general network matrix, not necessarily a doubly stochastic matrix.

C. Notation

Support-set \mathcal{T} of $\mathbf{x} = [x_1 x_2 \dots]^\top$ is defined as $\mathcal{T} = \{i : x_i \neq 0\}$. We use $|\mathcal{T}|$ and \mathcal{T}^c to denote the cardinality and complement of the set \mathcal{T} , respectively. For a 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

Algorithm 1 Distributed HTP - at node l

Input: $\mathbf{y}_l, \mathbf{A}_l, s, \{h_{lr}\}$

Initialization: $k \leftarrow 0$; $\hat{\mathbf{x}}_{l,k} \leftarrow \mathbf{0}$ (Estimate at k 'th iteration)

Iteration:

repeat

$k \leftarrow k + 1$ (Iteration counter)

1: $\tilde{\mathcal{T}}_{l,k} \leftarrow \text{supp}(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}), s)$

2: $\tilde{\mathbf{x}}_{l,k}$ such that $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}} \leftarrow \mathbf{A}_{l,\tilde{\mathcal{T}}_{l,k}}^\dagger \mathbf{y}_l$; $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$

3: $\tilde{\mathbf{x}}_{l,k} = \sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{r,k}$

4: $\hat{\mathcal{T}}_{l,k} \leftarrow \text{supp}(\tilde{\mathbf{x}}_{l,k}, s)$

5: $\hat{\mathbf{x}}_{l,k}$ such that $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}} \leftarrow \tilde{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}}$; $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$

until stopping criterion

Output: $\hat{\mathbf{x}}_l, \hat{\mathcal{T}}_l$

of \mathbf{x} indexed by $i \in \mathcal{T}$. Also we denote $(\cdot)^\top$ and $(\cdot)^\dagger$ as transpose and pseudo-inverse, respectively. We define the function $\text{supp}(\mathbf{x}, s) \triangleq \{\text{the set of indices corresponding to the } s \text{ largest amplitude components of } \mathbf{x}\}$. We use $\hat{\mathbf{x}}_{l,k}$ to denote the signal estimate at node l and iteration k .

II. DHTP ALGORITHM AND THEORETICAL ANALYSIS

The pseudo-code of the DHTP algorithm is shown in Algorithm 1. In every iteration k , the nodes use standard algorithmic steps of Hard Thresholding Pursuit algorithm (HTP) [35] along-with an extra step to include information about the estimates at the neighbors to refine the local estimate (see Step 3 of Algorithm 1). Here, we denote the neighborhood of node l by \mathcal{N}_l , i.e. $\mathcal{N}_l \triangleq \{r : h_{lr} \neq 0\}$. For theoretical analysis, we use the standard definition of RIP of a matrix as given in [36]. We denote the s -Restricted Isometry Constant (RIC) of a matrix by δ_s . We use $\|\cdot\|$ and $\|\cdot\|_0$ to denote the standard ℓ_2 and ℓ_0 norm of a vector, respectively. Throughout the paper unless specified, we have the following assumptions.

Assumption 1: The network matrix \mathbf{H} is a right stochastic matrix. This assumption is quite general as any non-negative matrix can be reduced to a right stochastic matrix by appropriately scaling the rows of the matrix.

Assumption 2: The sparsity level of the signal \mathbf{x} , denoted by $s \triangleq \|\mathbf{x}\|_0$ is known a-priori. This assumption is used in the greedy algorithms such as CoSaMP[8], subspace pursuit [7], HTP [35], etc.

We first provide a recurrence inequality for DHTP, which provides performance bounds of the algorithm over iterations. For notational clarity, we define RIC constant

$$\delta_{as} \triangleq \max_l \{\delta_{as}(\mathbf{A}_l)\},$$

where $\delta_{as}(\mathbf{A}_l)$ is the RIC of \mathbf{A}_l and a is a positive integer such as 1, 2 or 3.

Theorem 1 (Recurrence inequality): The performance of the DHTP algorithm at iteration k can be bounded as

$$\sum_{l=1}^L \|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| \leq c_1 \sum_{l=1}^L w_l \|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + d_1 \sum_{l=1}^L w_l \|\mathbf{e}_l\|,$$

where $w_l = \sum_r h_{rl}$, $c_1 = \sqrt{\frac{8\delta_{3s}^2}{1-\delta_{2s}^2}}$, $d_1 = \frac{2\sqrt{2(1-\delta_{2s})} + 2\sqrt{1+\delta_s}}{1-\delta_{2s}}$.

■

Detailed proof of the above theorem is shown in Section V. We use some intermediate steps in the proof of theorem 1 for addressing convergence of DHTP. We show convergence by two alternative approaches, in the following two theorems.

Theorem 2 (Convergence): Let x_j^* denote the magnitude of the j 'th highest amplitude element of \mathbf{x} and $\|\mathbf{e}\|_{\max} \triangleq \max_l \|\mathbf{e}_l\|$. If $\delta_{3s} < 1/3$ and $\|\mathbf{e}\|_{\max} \leq \gamma x_s^*$, then the DHTP algorithm converges after $\bar{k} = cs$ iterations, and its performance is bounded by

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,\bar{k}}\| \leq d \|\mathbf{e}\|_{\max},$$

where $c = \frac{\log(16c_3^2/c_1^4)}{\log(1/c_1^2)}$, $c_3 = \sqrt{\frac{16\delta_{3s}^2}{(1-\delta_{3s}^2)^2}}$, $d = \frac{4}{\sqrt{1-\delta_{3s}^2}}$ and $\gamma < 1$ are positive constants in terms of δ_{3s} . Under the above conditions, estimated support sets across all nodes are equal to the correct support set, that means

$$\forall l, \hat{\mathcal{T}}_l = \text{supp}(\mathbf{x}, s).$$

For an interpretation of the above theorem, we provide a numerical example. If $\delta_{3s} \leq 0.2$ then we have $c \leq 5$, $d \leq 4.47$; for an appropriate γ such that $\|\mathbf{e}\|_{\max} \leq \gamma x_s^*$, the performance $\|\mathbf{x} - \hat{\mathbf{x}}_{l,\bar{k}}\|$ is upper bounded by $4.47\|\mathbf{e}\|_{\max}$ after $5s$ iterations.

Corollary 1: Consider the special case of a doubly stochastic network matrix \mathbf{H} . Under the same conditions stated in Theorem 2, we have

$$\|\underline{\mathbf{x}} - \underline{\hat{\mathbf{x}}}_{\bar{k}}\| \leq d \|\underline{\mathbf{e}}\|,$$

where $\underline{\mathbf{x}} = [\mathbf{x}^\top \dots \mathbf{x}^\top]^\top$, $\underline{\hat{\mathbf{x}}}_{\bar{k}} = [\hat{\mathbf{x}}_{1,\bar{k}}^\top \dots \hat{\mathbf{x}}_{L,\bar{k}}^\top]^\top$ and $\underline{\mathbf{e}} = [\mathbf{e}_1^\top \dots \mathbf{e}_L^\top]^\top$. This upper bound is tighter than the bound of Theorem 2.

Theorem 3 (Convergence): If $\delta_{3s} < 1/3$ and $\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}} > 1$, then after $\bar{k} = \left\lceil \log\left(\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}}\right) / \log\left(\frac{1}{c_1}\right) \right\rceil$ iterations, DHTP algorithm converges and its performance is bounded by

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,\bar{k}}\| \leq d \|\mathbf{e}\|_{\max},$$

where $d = 1 + \frac{c_2 d_1}{1-c_1} + d_4$.

A relevant numerical example for interpretation of Theorem 3 is as follows: if $\delta_{3s} \leq 0.2$ and $\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}} = 20$ dB, then we have $\bar{k} = 9$ and $d = 13.68$. The proofs of the theorems and corollary are presented in Section V.

It can be seen that the DHTP algorithm has a convergence guarantee when $\delta_{3s} < 1/3$. Note that the requirement on signal-to-noise relation $\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}} > 1$ in Theorem 3 is weaker than the requirement $\|\mathbf{e}\|_{\max} \leq \gamma x_s^*$ in Theorem 2. The above results can be readily extended to the noiseless case, that means $\forall l, \mathbf{e}_l = \mathbf{0}$. For the noiseless case, DHTP provides the exact estimate of the sparse signal \mathbf{x} at every node.

A. Similarities and differences between DHTP and DiHaT

The DiHaT algorithm of [14] is shown in Algorithm 2. Comparing with Algorithm 1, the similarities and differences between DHTP and DiHaT are given in the list below.

- 1) DiHaT requires exchange of $\bar{\mathbf{y}}_{l,k}$, $\bar{\mathbf{A}}_{l,k}$ and $\tilde{\mathbf{x}}_{l,k}$ among nodes. DHTP requires only exchange of $\tilde{\mathbf{x}}_{l,k}$.

Algorithm 2 DiHaT - at node l

Input: $\mathbf{y}_l, \mathbf{A}_l, s, \{h_{lr}\}$

Initialization: $k \leftarrow 0$; $\hat{\mathbf{x}}_{l,0} \leftarrow \mathbf{0}$, $\bar{\mathbf{y}}_{l,0} \leftarrow \mathbf{y}_l$, $\bar{\mathbf{A}}_{l,0} \leftarrow \mathbf{A}_l$

Iteration:

repeat

$k \leftarrow k + 1$ (Iteration counter)

1: $\bar{\mathbf{y}}_{l,k} = \sum_{r \in \mathcal{N}_l} h_{lr} \bar{\mathbf{y}}_{r,k-1}$; $\bar{\mathbf{A}}_{l,k} = \sum_{r \in \mathcal{N}_l} h_{lr} \bar{\mathbf{A}}_{r,k-1}$

2: $\tilde{\mathcal{T}}_{l,k} \leftarrow \text{supp}(\hat{\mathbf{x}}_{l,k-1} + \bar{\mathbf{A}}_{l,k}^\top (\bar{\mathbf{y}}_{l,k} - \bar{\mathbf{A}}_{l,k} \hat{\mathbf{x}}_{l,k-1}), s)$

3: $\tilde{\mathbf{x}}_{l,k}$ such that $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}} \leftarrow (\bar{\mathbf{A}}_{l,k})_{\tilde{\mathcal{T}}_{l,k}}^\dagger \bar{\mathbf{y}}_{l,k}$; $\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$

4: $\hat{\mathbf{x}}_{l,k} = \sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{r,k}$

5: $\hat{\mathcal{T}}_{l,k} \leftarrow \text{supp}(\hat{\mathbf{x}}_{l,k}, s)$

6: $\hat{\mathbf{x}}_{l,k}$ such that $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}} \leftarrow \tilde{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}}$; $\hat{\mathbf{x}}_{\hat{\mathcal{T}}_{l,k}^c} \leftarrow \mathbf{0}$

until stopping criterion

Output: $\hat{\mathbf{x}}_l, \hat{\mathcal{T}}_l$

- 2) DiHaT requires \mathbf{H} to be a doubly stochastic matrix. This is not a requirement for the case of DHTP.
- 3) For theoretical convergence proof of DiHaT, an assumption is that the average noise over nodes $\frac{1}{L} \sum_l \mathbf{e}_l \rightarrow \mathbf{0}$. On the other hand, DHTP requires a signal-to-noise-ratio term $\frac{\|\mathbf{x}\|}{\|\mathbf{e}\|_{\max}} > 1$.
- 4) Denoting $\bar{\mathbf{A}} \triangleq \frac{1}{L} \sum_l \mathbf{A}_l$, DiHaT converges if $\delta_{3s}(\bar{\mathbf{A}}) < \frac{1}{3}$. On the other hand, DHTP converges if $\max_l \{\delta_{3s}(\mathbf{A}_l)\} < \frac{1}{3}$.
- 5) DiHaT provides consensus in the sense of achieving same estimation at all nodes under certain technical conditions. DHTP does not provide consensus except in the noiseless case under certain technical conditions.
- 6) DiHaT requires all \mathbf{A}_l to be of the same size. DHTP does not require this condition, that means dimension of \mathbf{y}_l can vary across nodes. *This is an important advantage in practical scenarios.*

III. SIMULATION RESULTS

In this section, we study the practical performance of DHTP using simulations and compare with DiHaT (Algorithm 2). We perform the study using Monte-Carlo simulations over many instances of \mathbf{A}_l , \mathbf{x} , and \mathbf{e}_l in the system model (1). The non-zero scalars of sparse signal are i.i.d. Gaussian. This is referred to as a Gaussian sparse signal. We set the number of nodes $L = 20$, and every node in the network is randomly connected to three other nodes apart from itself. The stopping criterion for the algorithms is that the maximum allowable number of iterations equal to 30. We used both right stochastic and doubly stochastic \mathbf{H} in simulations. Given an edge matrix of network connection between nodes, a right stochastic matrix generation is a simple task. The doubly stochastic matrix is generated through the *second largest eigenvalue modulus* (SLEM) optimization problem [37]. Finally, we also show performance for real image data.

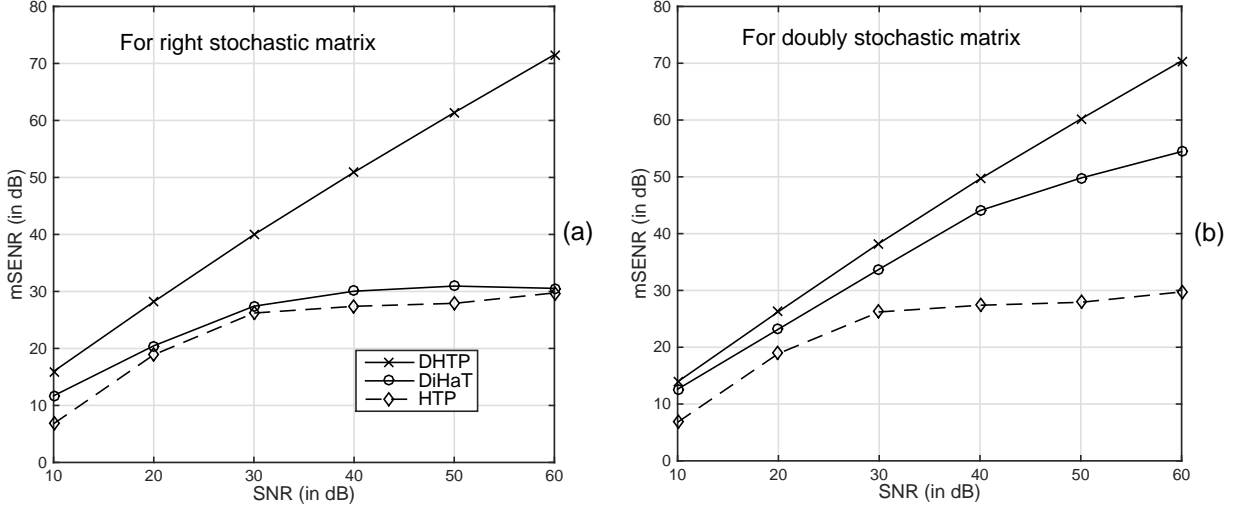


Fig. 1: mSENr performance of DHTP, DiHaT and HTP algorithms with respect to SNR. (a) Performance for a right stochastic network matrix. (b) Performance for a doubly stochastic network matrix.

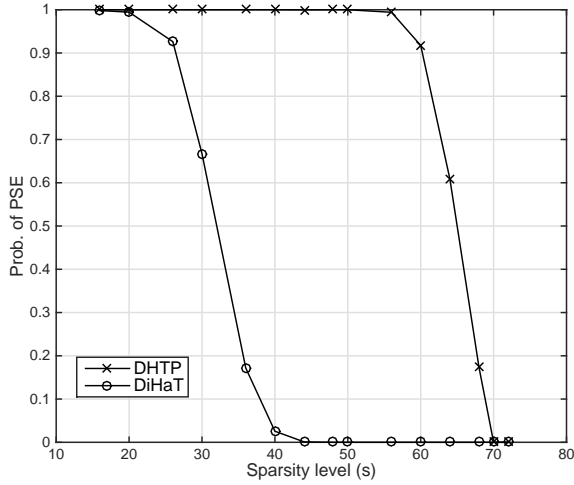


Fig. 2: Probability of perfect support-set estimation (PSE) versus sparsity level. No noise condition and we used a doubly stochastic network matrix.

A. Performance measures

For performance evaluation, we used a mean signal-to-estimation-noise ratio (mSENr) metric, $\text{mSENr} = \frac{1}{L} \sum_l \frac{\mathbb{E}\{\|\mathbf{x}_l\|^2\}}{\mathbb{E}\{\|\mathbf{x}_l - \hat{\mathbf{x}}_l\|^2\}}$. To generate noisy observations, we used Gaussian noise. The signal-to-noise ratio (SNR), $\text{SNR} = \text{SNR}_l = \frac{\mathbb{E}\{\|\mathbf{x}_l\|^2\}}{\mathbb{E}\{\|\mathbf{e}_l\|^2\}}$ is considered to be the same across all nodes.

B. Experiments using Simulated Data

We use all \mathbf{A}_l that have same row size, that is, $\forall l, M_l = M$. Same row size is necessary to use DiHaT for comparison. For the experiments, we set $M = 100$, and signal dimension $N = 500$. In our first experiment, we compare DHTP, DiHaT and HTP for right stochastic and doubly stochastic \mathbf{H} . The \mathbf{H} matrices are shown in the appendix. We set sparsity level $s = 20$. The results are shown in Fig. 1 where we show mSENr versus SNR. We recall that DiHaT was not designed for right stochastic \mathbf{H} and HTP is a standalone algorithm that

does not use the network. For right stochastic \mathbf{H} , we observe from Fig. 1 (a) that DiHaT does not provide considerable gain over HTP, but DHTP does. On the other hand, for doubly stochastic \mathbf{H} , we observe from Fig. 1 (b) that DiHaT provides a considerable gain over HTP, but DHTP outperforms DiHaT. The experiment validates that DHTP works for right stochastic \mathbf{H} . We did experiments with many instances of \mathbf{H} , and noted similar trend in performance.

Next, we study the probability of perfect signal estimation under a no-noise condition. Under this condition, the probability of perfect signal estimation is equivalent to the probability of perfect support-set estimation (PSE) at all nodes. Keeping $M = 100$ and $N = 500$ fixed, we vary the value of s and compute the probability of PSE using the frequentist approach – how many times PSE occurred. We used the same doubly stochastic \mathbf{H} of the first experiment. The result is shown in Fig. 2. It can be seen that the DHTP outperforms DiHaT in the sense of phase transition from perfect to imperfect estimation.

In the third experiment, we observe convergence speed of algorithms. A fast convergence leads to less usage of communication and computational resources, and less time delay in learning. We set $s = 20$. The results are shown in Fig. 3 where we show mSENr versus the number of iterations, for the noiseless condition and 30 dB SNR. We note that the DHTP has a significantly quicker convergence. In our experiments, the DHTP achieved convergence typically within five iterations.

Finally, we experiment to find the sensitivity of the DHTP and the DiHaT algorithms to the prior knowledge of sparsity level. For this, we use 30 dB SNR and $s = 20$. Fig. 4 shows the results for different assumed s that varies as $s = 18, 20, 25$, and 30. We observe that DHTP performs better than DiHaT for all assumed sparsity levels. Also, a typical trend is that the assumption of higher sparsity level is always better than lower sparsity level.

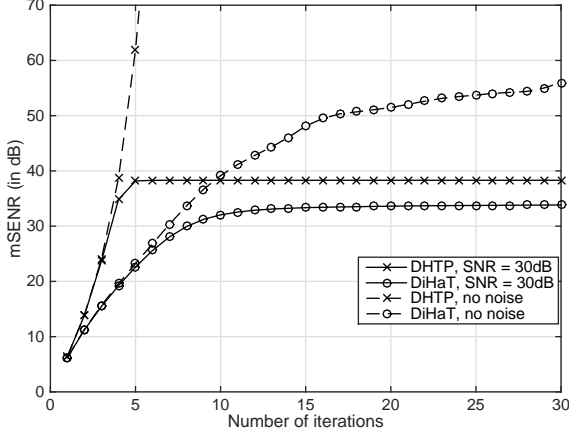


Fig. 3: mSENR performance of DHTP and DiHaT with respect to number of information exchange (number of iterations).

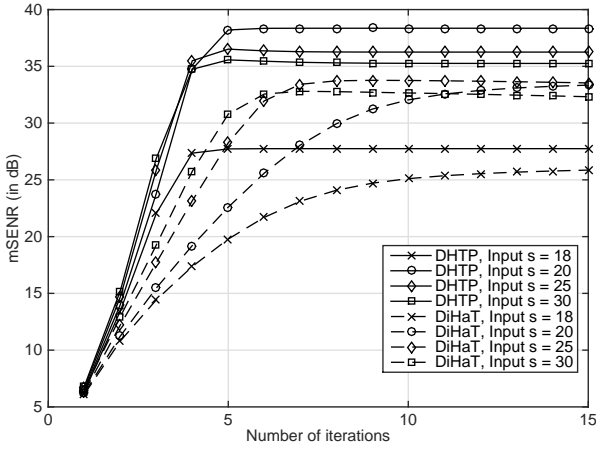


Fig. 4: Sensitivity performance of DHTP and DiHaT with respect to knowledge of the sparsity level. $M = 100$, $N = 500$, $s = 20$, $L = 20$, $d = 4$, $\text{SNR} = 30\text{dB}$.

C. Experiments for real data

We evaluate the performance on three standard grayscale images: Peppers, Lena and Baboon of size 512×512 pixels. We consider 11% of highest magnitude DCT coefficients of an image to decide a sparsity level choice. In DCT domain, the signal is split into 256 equal parts (or blocks) for ease of computation. This leads to the value of s for each part as close to 120. We perform reconstruction of the original images using the DHTP, DiHaT and HTP algorithms over the doubly stochastic network matrix \mathbf{H} chosen in the previous subsection. The performance measure is the peak-signal-to-noise-ratio (PSNR), defined as $\text{PSNR} = \frac{\|\mathbf{x}\|_\infty^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_\infty^2}$, where $\|\cdot\|_\infty$ denotes the ℓ_∞ norm. We show performance for a randomly chosen node among the set of 20 nodes. Fig. 5 shows a plot of the PSNR versus number of observations at each node (M). In the same figure, we also show visual reconstruction quality (reconstructed image) at $M = 240$ for DHTP. We observe that DHTP has a better convergence rate and PSNR performance than the other two algorithms.

D. Reproducible research

In the spirit of reproducible research, we provide relevant Matlab codes at www.ee.kth.se/reproducible/ and the link <https://sites.google.com/site/saikatcatt/software>. The code produces the results shown in the figures.

IV. CONCLUSION

For sparse learning over a network using distributed greedy algorithms such as the hard thresholding approach, we show that the strategy of exchanging signal estimates between nodes is good for learning. This has an explicit advantage of low communication overhead. We show that appropriate algorithmic strategies work for right stochastic network matrices. Use of right stochastic network matrices has higher generality than the popularly used doubly stochastic network matrices.

V. DETAILS OF THEORETICAL PROOFS

A. Useful Lemmas

We provide three lemmas here that will be used in the proofs later. The first lemma provides a bound for the orthogonal projection operation used in DHTP.

Lemma 1: [38, Lemma 2] Consider the standard sparse representation model $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ with $\|\mathbf{x}\|_0 = s_1$. Let $\mathcal{S} \subseteq \{1, 2, \dots, N\}$ and $|\mathcal{S}| = s_2$. Define $\bar{\mathbf{x}}$ such that $\bar{\mathbf{x}}_{\mathcal{S}} \leftarrow \mathbf{A}_{\mathcal{S}}^\dagger \mathbf{y}$; $\bar{\mathbf{x}}_{\mathcal{S}^c} \leftarrow \mathbf{0}$. If \mathbf{A} has RIC $\delta_{s_1+s_2} < 1$, then we have

$$\|\mathbf{x} - \bar{\mathbf{x}}\| \leq \sqrt{\frac{1}{1 - \delta_{s_1+s_2}^2}} \|\mathbf{x}_{\mathcal{S}^c}\| + \frac{\sqrt{1 + \delta_{s_2}}}{1 - \delta_{s_1+s_2}} \|\mathbf{e}\|.$$

The next lemma gives a useful inequality on squares of polynomials commonly encountered in the proofs.

Lemma 2: [38, Lemma 1] For non-negative numbers a, b, c, d, x, y ,

$$(ax + by)^2 + (cx + dy)^2 \leq \left(\sqrt{a^2 + c^2}x + (b + d)y \right)^2.$$

The last lemma provides a bound for the energy content in the pruned indices.

Lemma 3: [11, Lemma 3] Consider two vectors \mathbf{x} and \mathbf{z} with $\|\mathbf{x}\|_0 = s_1$, $\|\mathbf{z}\|_0 = s_2$ and $s_2 \geq s_1$. We have $\mathcal{S}_1 \triangleq \text{supp}(\mathbf{x}, s_1)$ and $\mathcal{S}_2 \triangleq \text{supp}(\mathbf{z}, s_2)$. Let \mathcal{S}_∇ denote the set of indices of the $s_2 - s_1$ smallest magnitude elements in \mathbf{z} . Then,

$$\|\mathbf{x}_{\mathcal{S}_\nabla}\| \leq \sqrt{2} \|(\mathbf{x} - \mathbf{z})_{\mathcal{S}_2}\| \leq \sqrt{2} \|\mathbf{x} - \mathbf{z}\|.$$

B. Proof of Theorem 1

At Step 2, using Lemma 1, we have

$$\begin{aligned} \|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\| &\leq \sqrt{\frac{1}{1 - \delta_{2s}^2}} \|\mathbf{x}_{\tilde{\mathcal{T}}_{l,k}^c}\| + \frac{\sqrt{1 + \delta_s}}{1 - \delta_{2s}} \|\mathbf{e}_l\|, \\ &\stackrel{(a)}{=} \sqrt{\frac{1}{1 - \delta_{2s}^2}} \|(\mathbf{x} - \tilde{\mathbf{x}}_{l,k})_{\tilde{\mathcal{T}}_{l,k}^c}\| + \frac{\sqrt{1 + \delta_s}}{1 - \delta_{2s}} \|\mathbf{e}_l\|, \end{aligned} \quad (2)$$

where (a) follows from the construction of $\tilde{\mathbf{x}}_{l,k}$. Following the proof of [35, Theorem 3.8], we can write,

$$\|(\mathbf{x} - \tilde{\mathbf{x}}_{l,k})_{\tilde{\mathcal{T}}_{l,k}^c}\| \leq \sqrt{2} \delta_{3s} \|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + \sqrt{2(1 + \delta_{2s})} \|\mathbf{e}_l\|. \quad (3)$$

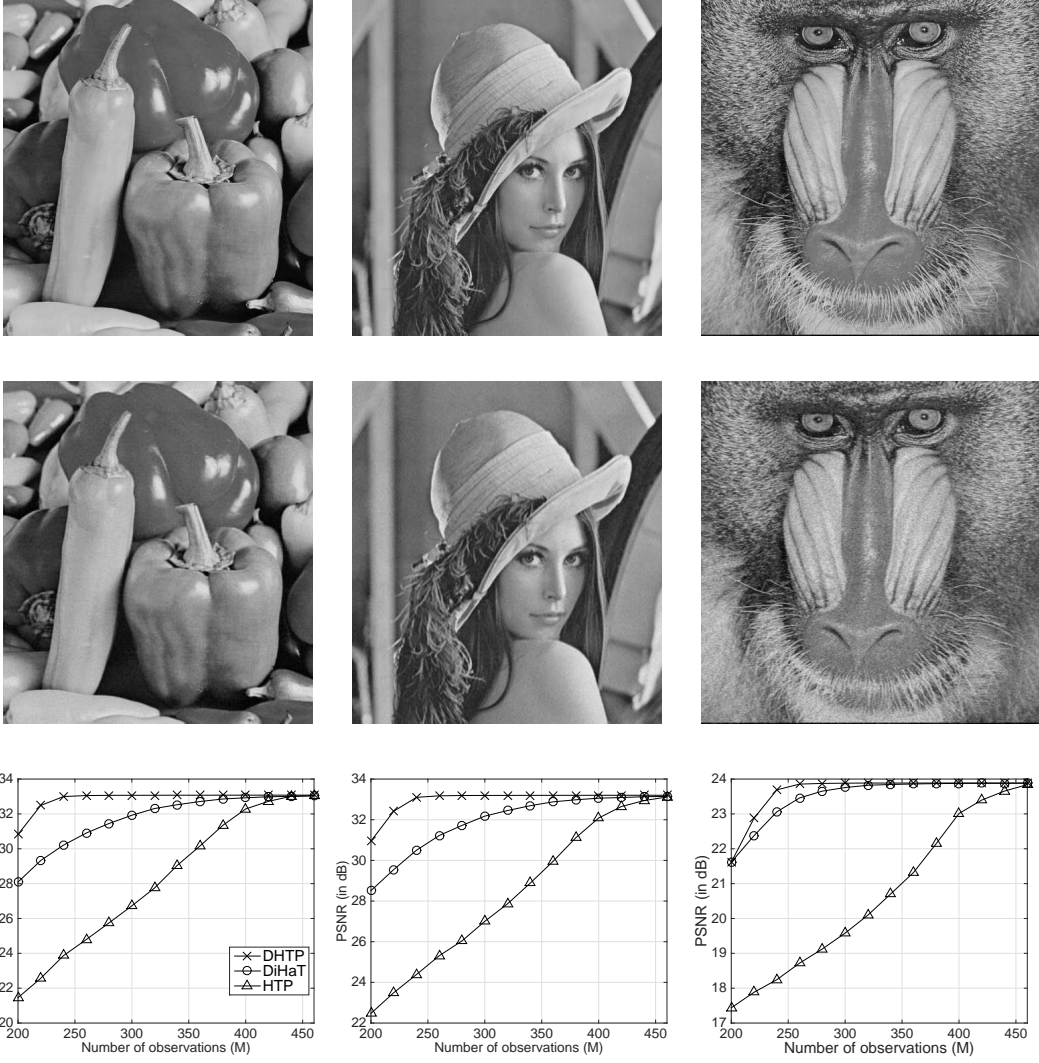


Fig. 5: Performance comparison of DHTP, DiHaT and HTP over image data. The first column, second column and the third column of the figure correspond to DHTP, DiHaT and HTP, respectively. The top row of the figure contains the original images. The second row contains the reconstructed images using DHTP with sparsity level 11%. The last row shows the PSNR performance of algorithms with respect to varying number of observations (M).

Substituting the above equation in (2), we have

$$\|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\| \leq \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}} \|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + \frac{d_1}{2} \|\mathbf{e}_l\|. \quad (4)$$

where $d_1 = \frac{2\sqrt{2(1-\delta_{2s})} + 2\sqrt{(1+\delta_s)}}{1-\delta_{2s}}$. Next, in step 3, we get

$$\begin{aligned} \|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\| &\stackrel{(a)}{=} \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \mathbf{x} - \sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{r,k} \right\| \\ &\stackrel{(b)}{\leq} \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \tilde{\mathbf{x}}_{r,k}\|, \end{aligned} \quad (5)$$

where (a) follows as $\sum_{r \in \mathcal{N}_l} h_{lr} = 1$ and (b) follows from the fact that h_{lr} is non-negative. Now, we bound the performance over the pruning step in steps 4-5 as follows,

$$\begin{aligned} \|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| &= \|(\mathbf{x} - \tilde{\mathbf{x}}_{l,k}) + (\tilde{\mathbf{x}}_{l,k} - \hat{\mathbf{x}}_{l,k})\| \\ &\stackrel{(a)}{\leq} \|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\| + \|\tilde{\mathbf{x}}_{l,k} - \hat{\mathbf{x}}_{l,k}\| \stackrel{(b)}{\leq} 2\|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\|, \end{aligned} \quad (6)$$

where (a) follows from the triangle inequality and (b) follows from the fact that $\hat{\mathbf{x}}_{l,k}$ is the best s -size approximation to $\tilde{\mathbf{x}}_{l,k}$. Combining (4), (5) and (6), we get

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| \leq c_1 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + d_1 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{e}_r\|.$$

Summing the above equation $\forall l$ and denoting $w_l = \sum_r h_{lr}$, we get the result of Theorem 1.

C. Proof of Theorem 2

Let π be the permutation of indices of \mathbf{x} such that $|x_{\pi(j)}| = x_j^*$ where $x_i^* \geq x_j^*$ for $i \leq j$. In other words, \mathbf{x}^* is sorted \mathbf{x} in the descending order of magnitude. Assuming $\pi(\{1, 2, \dots, p\}) \subseteq \hat{\mathcal{T}}_{l,k_1}$, we need to find the condition such that $\pi(\{1, 2, \dots, p+q\}) \subseteq \hat{\mathcal{T}}_{l,k_2}$ where $k_2 > k_1$. First, we have the following corollary.

Corollary 2:

$$\|\mathbf{x}_{\tilde{\mathcal{T}}_k^c}\| \leq c_1 \mathbf{H} \|\mathbf{x}_{\tilde{\mathcal{T}}_{k-1}^c}\| + (d_2 \mathbf{H} + d_3 \mathbf{I}) \|\mathbf{e}\|,$$

where $d_2 = \frac{2\delta_{3s}\sqrt{2(1+\delta_s)}}{1-\delta_{2s}}$, $d_3 = \sqrt{2(1+\delta_{2s})}$, $\|\mathbf{x}_{\tilde{\mathcal{T}}_k^c}\| = [\|\mathbf{x}_{\tilde{\mathcal{T}}_{1,k}^c}\| \dots \|\mathbf{x}_{\tilde{\mathcal{T}}_{L,k}^c}\|]^\top$ and $\|\mathbf{e}\| = [\|\mathbf{e}_1\| \dots \|\mathbf{e}_L\|]^\top$.

Proof The proof of the corollary follows from the following arguments. From (3), we can write

$$\|\mathbf{x}_{\tilde{\mathcal{T}}_{l,k}^c}\| \leq \sqrt{2}\delta_{3s}\|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + \sqrt{2(1+\delta_{2s})}\|\mathbf{e}_l\|,$$

due to the construction of $\tilde{\mathbf{x}}_{l,k}$. Substituting (2), (5) and (6) with ' $k-1$ ' in the above equation, we get

$$\|\mathbf{x}_{\tilde{\mathcal{T}}_{l,k}^c}\| \leq c_1 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x}_{\tilde{\mathcal{T}}_{r,k-1}^c}\| + d_2 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{e}_r\| + d_3 \|\mathbf{e}_l\|.$$

The result follows from vectorizing the above equation.

Next, Lemma 4 derives the condition that $\pi(\{1, 2, \dots, p+q\}) \subseteq \tilde{\mathcal{T}}_{l,k}$, i.e., the desired indices are selected in Step 2 of DHTP. Note that we define $\|\mathbf{e}\|_{\max} = [\|\mathbf{e}\|_{\max} \dots \|\mathbf{e}\|_{\max}]^\top$.

Lemma 4: If $\forall l, \pi(\{1, 2, \dots, p\}) \subset \tilde{\mathcal{T}}_{l,k_1}$ and

$$\mathbf{x}_{p+q}^* > c_1^{k_2-k_1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \frac{d_2+d_3}{1-c_1} \|\mathbf{e}\|_{\max}.$$

then, $\tilde{\mathcal{T}}_{l,k_2} \forall l$ contains $\pi(\{1, 2, \dots, p+q\})$.

Proof The first part of the proof of this lemma is similar to the proof of [39, Lemma 3]. It is enough to prove that the s highest magnitude indices of $(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))$ contains the indices $\pi(j)$ for $j \in \{1, 2, \dots, p+q\}$. Mathematically, we need to prove,

$$\min_{j \in \{1, 2, \dots, p+q\}} \|(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_{\pi(j)}\| > \max_{d \in T^c} \|(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_d\|, \forall l. \quad (7)$$

The LHS of (7) can be written as

$$\begin{aligned} & |(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_{\pi(j)}| \\ & \stackrel{(a)}{\geq} |\mathbf{x}_{\pi(j)}| - |(-\mathbf{x} + \hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_{\pi(j)}| \\ & \geq \mathbf{x}_{p+q}^* - |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}) + \mathbf{A}_l^\top \mathbf{e}_l)_{\pi(j)}|, \end{aligned}$$

where (a) follows from the reverse triangle inequality. Similarly, the RHS of (7) can be written as

$$\begin{aligned} & |(\hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_d| \\ & = |\mathbf{x}_d + (-\mathbf{x} + \hat{\mathbf{x}}_{l,k-1} + \mathbf{A}_l^\top (\mathbf{y}_l - \mathbf{A}_l \hat{\mathbf{x}}_{l,k-1}))_d| \\ & = |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}) + \mathbf{A}_l^\top \mathbf{e}_l)_d|. \end{aligned}$$

Using the bounds on LHS and RHS, (7) simplifies to

$$\mathbf{x}_{p+q}^* > |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}) + \mathbf{A}_l^\top \mathbf{e}_l)_{\pi(j)}| + |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}) + \mathbf{A}_l^\top \mathbf{e}_l)_d|.$$

Let RHS of the sufficient condition at node l be denoted as RHS_l . Then,

$$\begin{aligned} \text{RHS}_l & \leq \sqrt{2} |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}) + \mathbf{A}_l^\top \mathbf{e}_l)_{\{\pi(j), d\}}| \\ & \leq \sqrt{2} |((\mathbf{A}_l^\top \mathbf{A}_l - \mathbf{I})(\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}))_{\{\pi(j), d\}}| \\ & \quad + \sqrt{2} \|(\mathbf{A}_l^\top \mathbf{e}_l)_{\{\pi(j), d\}}\| \\ & \stackrel{(a)}{\leq} \sqrt{2}\delta_{3s} \|\mathbf{x} - \hat{\mathbf{x}}_{l,k-1}\| + \sqrt{2(1+\delta_{2s})} \|\mathbf{e}_l\| \\ & \stackrel{(b)}{\leq} c_1 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x}_{\tilde{\mathcal{T}}_{r,k-1}^c}\| + d_2 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{e}_r\| + d_3 \|\mathbf{e}_l\|, \end{aligned}$$

where (a) follows from [38, Lemma 4-5] and (b) follows from substituting (2), (5) and (6). At iteration k_2 , RHS_l can be vectorized as

$$\underline{\text{RHS}} = [\text{RHS}_1 \dots \text{RHS}_L]^\top \leq c_1 \mathbf{H} \|\mathbf{x}_{\tilde{\mathcal{T}}_{k_2-1}^c}\| + (d_2 \mathbf{H} + d_3 \mathbf{I}) \|\mathbf{e}\|.$$

Applying Corollary 2 repeatedly, we can write for $k_1 < k_2$,

$$\begin{aligned} \underline{\text{RHS}} & \leq (c_1 \mathbf{H})^{k_2-k_1} \|\mathbf{x}_{\tilde{\mathcal{T}}_{k_1}^c}\| \\ & \quad + (d_2 \mathbf{H} + d_3 \mathbf{I}_L) (\mathbf{I}_L + \dots + (c_1 \mathbf{H})^{k_2-k_1-1}) \|\mathbf{e}\| \\ & \stackrel{(a)}{\leq} c_1^{k_2-k_1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \frac{d_2+d_3}{1-c_1} \|\mathbf{e}\|_{\max}, \end{aligned}$$

where (a) follows from the assumption that for any l , $\|\mathbf{x}_{\tilde{\mathcal{T}}_{l,k_1}^c}\| \leq \|\mathbf{x}_{\{p+1, \dots, s\}}^*\|$ and the right stochastic property of \mathbf{H} . Now, it can be seen that the bound in (7) is satisfied when

$$\mathbf{x}_{p+q}^* > c_1^{k_2-k_1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \frac{d_2+d_3}{1-c_1} \|\mathbf{e}\|_{\max}.$$

Next, we find the condition that $\pi(\{1, 2, \dots, p+q\}) \subseteq \tilde{\mathcal{T}}_{l,k}$ in the following Lemma.

Lemma 5: If $\pi(\{1, 2, \dots, p\}) \subseteq \tilde{\mathcal{T}}_{l,k_1} \forall l$, $\pi(\{1, 2, \dots, p+q\}) \subseteq \tilde{\mathcal{T}}_{l,k_2} \forall l$ and

$$\mathbf{x}_{p+q}^* > c_3 c_1^{k_2-k_1-1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \left(\frac{c_3(d_2+d_3)}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}$$

then, $\tilde{\mathcal{T}}_{l,k_2} \forall l$ contains $\pi(\{1, 2, \dots, p+q\})$. The constant $d_4 = \frac{2d_2}{\sqrt{1-\delta_{3s}}} + \frac{d_1}{\sqrt{2}}$.

Proof It is enough to prove that the s highest magnitude indices of $\tilde{\mathbf{x}}_{l,k}$ contains the indices $\pi(j)$ for $j \in \{1, 2, \dots, p+q\}$. Mathematically, we need to prove,

$$\min_{j \in \{1, 2, \dots, p+q\}} \|(\tilde{\mathbf{x}}_{l,k})_{\pi(j)}\| > \max_{d \in T^c} \|(\tilde{\mathbf{x}}_{l,k})_d\|, \forall l. \quad (8)$$

The LHS of (8) can be written as

$$\begin{aligned} \|(\tilde{\mathbf{x}}_{l,k})_{\pi(j)}\| & \stackrel{(a)}{=} \left\| \left(\sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} \right)_{\pi(j)} \right\| \\ & \stackrel{(b)}{=} \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \mathbf{x}_{\pi(j)} + \sum_{r \in \mathcal{N}_l} h_{lr} (\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}})_{\pi(j)} \right\| \\ & \geq \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \mathbf{x}_{\pi(j)} \right\| - \left\| \sum_{r \in \mathcal{N}_l} h_{lr} (\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}})_{\pi(j)} \right\| \\ & \geq \mathbf{x}_{p+q}^* - \left\| \sum_{r \in \mathcal{N}_l} h_{lr} (\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}})_{\pi(j)} \right\|, \end{aligned}$$

where (a) and (b) follows from the fact that $\pi(\{1, 2, \dots, p+q\}) \subset \tilde{\mathcal{T}}_{l,k_2} \forall l$. Similarly, the RHS of (8) can be bounded as

$$\begin{aligned} \|(\tilde{\mathbf{x}}_{l,k})_d\| & = \left\| \left(\sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} \right)_d \right\| \\ & = \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \mathbf{x}_d + \sum_{r \in \mathcal{N}_l} h_{lr} (\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}})_d \right\| \\ & = \left\| \sum_{r \in \mathcal{N}_l} h_{lr} (\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}})_d \right\|. \end{aligned}$$

Using the above two bounds, the condition (8) can now be written as

$$\mathbf{x}_{p+q}^* > \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \left(\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}} \right)_{\pi(j)} \right\| + \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \left(\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}} \right)_d \right\|. \quad (9)$$

Define the RHS of the required condition from (9) at node l as RHS_l . Then, we can write the sufficient condition as

$$\begin{aligned} \text{RHS}_l &\leq \sqrt{2} \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \left(\tilde{\mathbf{x}}_{\tilde{\mathcal{T}}_{r,k}} - \mathbf{x}_{\tilde{\mathcal{T}}_{r,k}} \right)_{\{\pi(j), d\}} \right\| \\ &\leq \sqrt{2} \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \tilde{\mathbf{x}}_{r,k}\|. \end{aligned}$$

From the above equation and (4), we can bound RHS_l as

$$\begin{aligned} \text{RHS}_l &\leq \sum_{r \in \mathcal{N}_l} h_{lr} \left(\sqrt{\frac{4\delta_{3s}^2}{1-\delta_{3s}}} \|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + \frac{d_1}{\sqrt{2}} \|\mathbf{e}_r\| \right) \\ &\stackrel{(a)}{\leq} c_3 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x}_{\tilde{\mathcal{T}}_{r,k-1}^c}\| + d_4 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{e}_r\|, \end{aligned}$$

where (a) follows from substituting (2), (5) and (6). At iteration k_2 , RHS_l can be vectorized as

$$\underline{\text{RHS}} = [\text{RHS}_1 \dots \text{RHS}_L]^\top \leq c_3 \underline{\mathbf{H}} \|\mathbf{x}_{\tilde{\mathcal{T}}_{k_2-1}^c}\| + d_4 \underline{\mathbf{H}} \|\mathbf{e}\|.$$

Applying Corollary 2 repeatedly, we can write for $k_1 < k_2$,

$$\begin{aligned} \underline{\text{RHS}} &\leq c_3 (c_1 \underline{\mathbf{H}})^{k_2-k_1-1} \underline{\mathbf{H}} \|\mathbf{x}_{\tilde{\mathcal{T}}_{k_1}^c}\| \\ &+ (c_3 (d_2 \underline{\mathbf{H}} + d_3 \underline{\mathbf{I}}) (\underline{\mathbf{I}}_L + \dots + (c_1 \underline{\mathbf{H}})^{k_2-k_1-2}) + d_4) \underline{\mathbf{H}} \|\mathbf{e}\| \\ &\stackrel{(a)}{\leq} c_3 c_1^{k_2-k_1-1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \left(\frac{c_3 (d_2 + d_3)}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}, \end{aligned}$$

where (a) follows from the assumption that for any l , $\|\mathbf{x}_{\tilde{\mathcal{T}}_{l,k_1}^c}\| \leq \|\mathbf{x}_{\{p+1, \dots, s\}}^*\|$ and the right stochastic property of $\underline{\mathbf{H}}$. From the above bound on $\underline{\text{RHS}}$, it can be easily seen that (9) is satisfied when

$$\mathbf{x}_{p+q}^* > c_3 c_1^{k_2-k_1-1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \left(\frac{c_3 (d_2 + d_3)}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}.$$

Corollary 3: For $\delta_{3s} < 1$, we have $c_1 < c_3$. Therefore, the sufficient condition for Lemma 4 and Lemma 5 to hold is $\mathbf{x}_{p+q}^* > c_3 c_1^{k_2-k_1-1} \|\mathbf{x}_{\{p+1, \dots, s\}}^*\| + \left(\frac{c_3 (d_2 + d_3)}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}$.

Next, the above corollary can be used to prove theorem 2 by similar steps as outlined in [39, Theorem 6]. In the proof, the number of iterations k_i between different steps for our case should be defined as $k_i := \left\lceil \frac{\log(16(c_3/c_1)^2(|Q_i|+|Q_{i+1}|/2+\dots+|Q_r|/2^{r-i}))}{\log(1/c_1^2)} \right\rceil$. All the variables in the above definition are defined in [39, Theorem 6] except c_1, c_3 which are defined in this paper. With the above definition, the number of iterations is bounded as $\bar{k} = cs$. The proof steps in [39, Theorem 6] require that $c_1 < 1$ which holds as $\delta_{3s} < 0.362$.

Note that the constant γ is defined as $\gamma = \frac{2\sqrt{2}-1}{4d_5}$, where $d_5 = \frac{c_3(d_2+d_3)}{1-c_1} + d_4$. The constant d in the theorem statement can be derived as follows. From Step 2 of DHTP at node l ,

we have $\|\mathbf{y}_l - \mathbf{A}_l \tilde{\mathbf{x}}_{l,\bar{k}}\| \leq \|\mathbf{y}_l - \mathbf{A}_l \mathbf{x}\| = \|\mathbf{e}_l\| \leq \|\mathbf{e}\|_{\max}$. This follows because $\tilde{\mathcal{T}}_{l,\bar{k}} = \mathcal{T}$. Next, we can write

$$\begin{aligned} &\|\mathbf{x} - \hat{\mathbf{x}}_{l,\bar{k}}\| \\ &\stackrel{(a)}{\leq} 2 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \tilde{\mathbf{x}}_{r,\bar{k}}\| \leq \sum_{r \in \mathcal{N}_l} h_{lr} \frac{2}{\sqrt{1-\delta_{2s}}} \|\mathbf{A}_r (\mathbf{x} - \tilde{\mathbf{x}}_{r,\bar{k}})\| \\ &\leq \sum_{r \in \mathcal{N}_l} h_{lr} \frac{2}{\sqrt{1-\delta_{2s}}} (\|\mathbf{y}_r - \mathbf{A}_r \tilde{\mathbf{x}}_{r,\bar{k}}\| + \|\mathbf{e}_r\|) \leq \frac{4}{\sqrt{1-\delta_{3s}}} \|\mathbf{e}\|_{\max}, \end{aligned}$$

where (a) follows from (5), (6) and the last inequality follows from the fact that $\delta_{2s} < \delta_{3s}$ and the right stochastic property of $\underline{\mathbf{H}}$. \square

D. Proof of Corollary 1

From Step 2 of DHTP, we have $\|\underline{\mathbf{y}} - \underline{\mathbf{A}} \underline{\tilde{\mathbf{x}}}_{\bar{k}}\|^2 \leq \|\underline{\mathbf{y}} - \underline{\mathbf{A}} \underline{\mathbf{x}}\|^2 = \|\underline{\mathbf{e}}\|^2$, where $\underline{\mathbf{y}} = [\mathbf{y}_1^\top \dots \mathbf{y}_L^\top]^\top$, $\underline{\tilde{\mathbf{x}}}_{\bar{k}} = [\tilde{\mathbf{x}}_{1,\bar{k}}^\top \dots \tilde{\mathbf{x}}_{L,\bar{k}}^\top]^\top$ and $\underline{\mathbf{A}} = \underline{\mathbf{I}}_L \otimes \mathbf{A}$. The first inequality follows because $\tilde{\mathcal{T}}_{l,\bar{k}} = \mathcal{T}, \forall l$. Define $\underline{\mathbf{H}} \triangleq \underline{\mathbf{H}} \otimes \underline{\mathbf{I}}_L$. Then, we have,

$$\begin{aligned} \|\underline{\mathbf{x}} - \underline{\tilde{\mathbf{x}}}_{\bar{k}}\|^2 &\stackrel{(a)}{\leq} 4 \|\underline{\mathbf{x}} - \underline{\tilde{\mathbf{x}}}_{\bar{k}}\|^2 \stackrel{(b)}{\leq} 4 \|\underline{\mathbf{H}} \underline{\mathbf{x}} - \underline{\mathbf{H}} \underline{\tilde{\mathbf{x}}}_{\bar{k}}\|^2 \stackrel{(c)}{\leq} 4 \|\underline{\mathbf{x}} - \underline{\tilde{\mathbf{x}}}_{\bar{k}}\|^2 \\ &\stackrel{(d)}{\leq} \frac{4}{1-\delta_{2s}} \left\| \underline{\mathbf{A}} (\underline{\mathbf{x}} - \underline{\tilde{\mathbf{x}}}_{\bar{k}}) \right\|^2 = \frac{4}{1-\delta_{2s}} \left\| \underline{\mathbf{y}} - \underline{\mathbf{e}} - \underline{\mathbf{A}} \underline{\tilde{\mathbf{x}}}_{\bar{k}} \right\|^2, \end{aligned}$$

where (a) follows from (6), and (b), (c) follows from the doubly stochastic property of $\underline{\mathbf{H}}$ ($\|\underline{\mathbf{H}}\| = 1$). Also, (d) follows from the RIP property of $\underline{\mathbf{A}}$. Further, we can write

$$\|\underline{\mathbf{x}} - \underline{\tilde{\mathbf{x}}}_{\bar{k}}\| \leq \frac{2}{\sqrt{1-\delta_{2s}}} (\|\underline{\mathbf{y}} - \underline{\mathbf{A}} \underline{\tilde{\mathbf{x}}}_{\bar{k}}\| + \|\underline{\mathbf{e}}\|) \leq \frac{4}{\sqrt{1-\delta_{3s}}} \|\underline{\mathbf{e}}\|,$$

where the last inequality follows from the fact that $\delta_{2s} < \delta_{3s}$.

E. Proof of Theorem 3

We define, $\nabla \hat{\mathcal{T}}_{l,k} \triangleq \left\{ \bigcup_{r \in \mathcal{N}_l} \tilde{\mathcal{T}}_{r,k} \right\} \setminus \hat{\mathcal{T}}_{l,k}$. Then, at step 5 of DHTP we have

$$\begin{aligned} \|\mathbf{x}_{\hat{\mathcal{T}}_{l,k}^c}\|^2 &= \|\mathbf{x}_{\nabla \hat{\mathcal{T}}_{l,k}}\|^2 + \|\mathbf{x}_{\left\{ \bigcap_{r \in \mathcal{N}_l} \tilde{\mathcal{T}}_{r,k} \right\}}\|^2 \\ &\stackrel{(a)}{\leq} \|\mathbf{x}_{\nabla \hat{\mathcal{T}}_{l,k}}\|^2 + \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x}_{\tilde{\mathcal{T}}_{r,k}^c}\|^2, \end{aligned} \quad (10)$$

where (a) follows from the right stochastic property of $\underline{\mathbf{H}}$. Also, from Lemma 3, we have

$$\begin{aligned} \|\mathbf{x}_{\nabla \hat{\mathcal{T}}_{l,k}}\| &\leq \sqrt{2} \|\mathbf{x} - \tilde{\mathbf{x}}_{l,k}\| \stackrel{(a)}{=} \sqrt{2} \left\| \sum_{r \in \mathcal{N}_l} h_{lr} \mathbf{x} - \sum_{r \in \mathcal{N}_l} h_{lr} \tilde{\mathbf{x}}_{r,k} \right\| \\ &\stackrel{(b)}{\leq} \sqrt{2} \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \tilde{\mathbf{x}}_{r,k}\|, \end{aligned} \quad (11)$$

where (a) and (b) follows from the assumption that $\forall l, \sum_{r \in \mathcal{N}_l} h_{lr} = 1$ and $\forall \{l, r\}$, the value, $h_{lr} \geq 0$ respectively.

Combining (10) and (11), we have

$$\begin{aligned} \|\mathbf{x}_{\hat{\mathcal{T}}_{l,k}^c}\|^2 &\stackrel{(3),(4)}{\leq} \left[\sum_{r \in \mathcal{N}_l} h_{lr} \left(\sqrt{\frac{4\delta_{3s}^2}{1-\delta_{3s}}} \|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + \frac{d_1}{\sqrt{2}} \|\mathbf{e}_r\| \right) \right]^2 \\ &+ \left[\sum_{r \in \mathcal{N}_l} h_{lr} \left(\sqrt{2}\delta_{3s} \|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + \sqrt{2(1+\delta_{2s})} \|\mathbf{e}_r\| \right) \right]^2. \end{aligned}$$

Using Lemma 2, the above equation can be simplified as

$$\|\mathbf{x}_{\hat{\mathcal{T}}_{l,k}^c}\| \leq c_2 \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| + \left(\frac{d_1}{\sqrt{2}} + d_3 \right) \sum_{r \in \mathcal{N}_l} h_{lr} \|\mathbf{e}_r\|,$$

where $c_2 = \sqrt{\frac{2\delta_{3s}^2(3-\delta_{3s}^2)}{1-\delta_{3s}^2}}$. This equation can be vectorized as

$$\|\mathbf{x}_{\hat{\mathcal{T}}_k^c}\| \leq c_2 \mathbf{H} \|\mathbf{x} - \hat{\mathbf{x}}_{k-1}\| + d_4 \|\mathbf{e}\|, \quad (12)$$

where $\|\mathbf{x} - \hat{\mathbf{x}}_{r,k-1}\| = [\|\mathbf{x} - \hat{\mathbf{x}}_{1,k-1}\| \dots \|\mathbf{x} - \hat{\mathbf{x}}_{L,k-1}\|]^\top$ and $d_4 = \frac{d_1}{\sqrt{2}} + d_3$. From the proof of Theorem 1, we have the following relation in vectorized form

$$\|\mathbf{x} - \hat{\mathbf{x}}_k\| \leq c_1 \mathbf{H} \|\mathbf{x} - \hat{\mathbf{x}}_{k-1}\| + d_1 \mathbf{H} \|\mathbf{e}\|.$$

Applying the above relation repeatedly, and using the fact that $c_1 < 1$ (as $\delta_{3s} < 0.362$), we can write (12) at the iteration \bar{k} as

$$\begin{aligned} \|\mathbf{x}_{\hat{\mathcal{T}}_k^c}\| &\leq c_2 (c_1 \mathbf{H})^{\bar{k}-1} \mathbf{H} \|\mathbf{x} - \hat{\mathbf{x}}_0\| \\ &\quad + \left(c_2 d_1 \mathbf{H} (\mathbf{I}_L + \dots + (c_1 \mathbf{H})^{\bar{k}-2}) + d_4 \right) \|\mathbf{e}\|_{\max} \\ &\stackrel{(a)}{\leq} c_1^{\bar{k}} \|\mathbf{x}\| + \left(\frac{c_2 d_1}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}, \end{aligned}$$

where (a) follows from the fact that $c_2 < c_1$, the initial condition, $\|\mathbf{x} - \hat{\mathbf{x}}_{l,0}\| = \|\mathbf{x}\|, \forall l$ and the right stochastic property of \mathbf{H} . We have also defined, $\|\mathbf{x}\| = [\|\mathbf{x}\| \dots \|\mathbf{x}\|]^\top$. Substituting the value of \bar{k} in the above equation, we get

$$\|\mathbf{x}_{\hat{\mathcal{T}}_k^c}\| \leq \left(1 + \frac{c_2 d_1}{1-c_1} + d_4 \right) \|\mathbf{e}\|_{\max}.$$

□

REFERENCES

- [1] M. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learning Res.*, vol. 1, pp. 211–244, 2001.
- [2] J. Wright, A. Yang, A. Ganesh, S. Sastry, and M. Yi, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 210–227, Feb. 2009.
- [3] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*. Springer, 2010.
- [4] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Mag.*, vol. 25, pp. 92–101, Mar. 2008.
- [5] H. Schepker and A. Dekorsy, "Sparse multi-user detection for cdma transmission using greedy algorithms," in *Int. Symp. on Wireless Commun. Sys. (ISWCS)*, pp. 291–295, Nov. 2011.
- [6] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.
- [7] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, pp. 2230–2249, May 2009.
- [8] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Tech. Rep., California Inst. Tech.*, Jul. 2008.
- [9] H. Huang and A. Makur, "Backtracking-based matching pursuit method for sparse signal reconstruction," *IEEE Signal Process. Lett.*, vol. 18, pp. 391–394, Jul. 2011.
- [10] S. Chatterjee, D. Sundman, M. Vehkaperä, and M. Skoglund, "Projection-based and look-ahead strategies for atom selection," *IEEE Trans. Signal Process.*, vol. 60, pp. 634–647, Feb. 2012.
- [11] A. Zaki, A. Venkitaraman, S. Chatterjee, and L. K. Rasmussen, "Greedy sparse learning over network," *IEEE Trans. Signal and Inf. Process. over Networks*, vol. PP, no. 99, pp. 1–1, 2017.
- [12] D. Sundman, S. Chatterjee, and M. Skoglund, "Design and analysis of a greedy pursuit for distributed compressed sensing," *IEEE Transactions on Signal Processing*, vol. 64, pp. 2803–2818, June 2016.
- [13] D. Sundman, S. Chatterjee, and M. Skoglund, "Distributed greedy pursuit algorithms," *Signal Processing*, vol. 105, no. 0, pp. 298–315, 2014.
- [14] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "Greedy sparsity-promoting algorithms for distributed learning," *IEEE Trans. Signal Process.*, vol. 63, pp. 1419–1432, Mar. 2015.
- [15] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, pp. 4931–4946, Oct. 2014.
- [16] P. Han, R. Niu, and Y. C. Eldar, "Modified distributed iterative hard thresholding," in *2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3766–3770, Apr. 2015.
- [17] T. Wimalajeewa and P. K. Varshney, "Omp based joint sparsity pattern recovery under communication constraints," *IEEE Trans. Signal Process.*, vol. 62, pp. 5059–5072, Oct. 2014.
- [18] D. Sundman, S. Chatterjee, and M. Skoglund, "Design and analysis of a greedy pursuit for distributed compressed sensing," *IEEE Trans. Signal Process.*, vol. 64, pp. 2803–2818, Jun. 2016.
- [19] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Thirty-Ninth Asilomar Conf. Signals, Systems and Computers*, pp. 1537–1541, Nov. 2005.
- [20] D. Baron, M. Duarte, M. Wakin, S. Sarvotham, and R. Baraniuk, "Distributed compressive sensing," Jan. 2009.
- [21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.
- [22] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, pp. 1942–1956, Apr. 2012.
- [23] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, pp. 5262–5276, Oct. 2010.
- [24] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, pp. 5412–5425, Oct. 2012.
- [25] Z. Liu, Y. Liu, and C. Li, "Distributed sparse recursive least-squares over networks," *IEEE Trans. Signal Process.*, vol. 62, pp. 1386–1395, Mar. 2014.
- [26] P. D. Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, pp. 1419–1433, Mar. 2013.
- [27] P. Han, R. Niu, M. Ren, and Y. C. Eldar, "Distributed approximate message passing for sparse signal recovery," in *2014 IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, pp. 497–501, Dec. 2014.
- [28] M. Masood and T. Y. Al-Naffouri, "Sparse reconstruction using distribution agnostic bayesian matching pursuit," *IEEE Trans. Signal Process.*, vol. 61, pp. 5298–5309, Nov. 2013.
- [29] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, pp. 349–396, Feb. 2003.
- [30] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Trans. Image Process.*, vol. 23, pp. 2528–2541, Jun. 2014.
- [31] T. Wimalajeewa and P. K. Varshney, "Wireless compressive sensing over fading channels with distributed sparse random projections," *IEEE Trans. Signal and Inf. Process. over Networks*, vol. 1, pp. 33–44, Mar. 2015.
- [32] S. M. Fosson, J. Matamoros, C. Antn-Haro, and E. Magli, "Distributed recovery of jointly sparse signals under communication constraints," *IEEE Trans. Signal Process.*, vol. 64, pp. 3470–3482, Jul. 2016.
- [33] J. Matamoros, S. M. Fosson, E. Magli, and C. Antn-Haro, "Distributed admm for in-network reconstruction of sparse signals with innovations," *IEEE Trans. Signal and Inf. Process. over Networks*, vol. 1, pp. 225–234, Dec. 2015.
- [34] W. Chen and I. J. Wassell, "A decentralized bayesian algorithm for distributed compressive sensing in networked sensing systems," *IEEE Trans. Wireless Commun.*, vol. 15, pp. 1282–1292, Feb. 2016.
- [35] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM Journal on Numerical Analysis*, vol. 49, no. 6, pp. 2543–2563, 2011.
- [36] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.
- [37] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Rev.*, vol. 46, pp. 667–689, Apr. 2004.
- [38] C. B. Song, S. T. Xia, and X. J. Liu, "Improved analysis for subspace pursuit algorithm in terms of restricted isometry constant," *IEEE Signal Process. Lett.*, vol. 21, pp. 1365–1369, Nov. 2014.
- [39] J.-L. Bouchot, S. Foucart, and P. Hitzchenko, "Hard thresholding pursuit algorithms: Number of iterations," *Applied and Computational Harmonic Analysis*, vol. 41, pp. 412–435, Sep. 2016.

APPENDIX

A. Right stochastic matrix used in simulation experiment

$$\mathbf{H} = \begin{bmatrix} 0.28 & 0 & 0 & 0 & 0 & 0 & 0.24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.30 & 0 & 0.18 \\ 0 & 0.25 & 0 & 0 & 0 & 0.16 & 0.24 & 0 & 0.35 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.32 & 0 & 0.14 & 0 & 0 & 0 & 0 & 0 & 0.16 & 0 & 0 & 0.38 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.28 & 0 & 0 & 0 & 0 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0 & 0.31 & 0 & 0 & 0 \\ 0 & 0 & 0.18 & 0 & 0.28 & 0 & 0.35 & 0 & 0 & 0.19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.19 & 0 & 0 & 0 & 0.12 & 0 & 0 & 0 & 0 & 0 & 0 & 0.28 & 0 & 0.41 & 0 & 0 & 0 & 0 & 0 \\ 0.24 & 0.25 & 0 & 0 & 0.28 & 0 & 0.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.26 & 0 & 0 & 0 & 0.12 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0.40 & 0 & 0 & 0 & 0 \\ 0 & 0.33 & 0 & 0 & 0 & 0 & 0 & 0.24 & 0.32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.04 & 0 & 0 & 0 & 0 & 0.37 & 0 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0 & 0 & 0.40 \\ 0 & 0 & 0.27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.09 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0.08 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.20 & 0 & 0 & 0 & 0 & 0 & 0 & 0.22 & 0.31 & 0 & 0 & 0.27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.26 & 0.26 & 0 & 0 & 0 & 0.26 & 0 & 0 & 0 \\ 0 & 0 & 0.06 & 0 & 0 & 0.31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.24 & 0 & 0 & 0 & 0 & 0.39 \\ 0 & 0 & 0 & 0.20 & 0 & 0 & 0 & 0.30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0.42 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.06 & 0.39 & 0.20 & 0 & 0.24 & 0.06 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.39 & 0 & 0.24 & 0 & 0 & 0 & 0 & 0 & 0.27 & 0 & 0.10 \\ 0.27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.24 & 0.23 & 0.26 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 & 0.16 & 0.23 & 0.32 \\ 0.10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.21 & 0 & 0 & 0.24 & 0.04 & 0.29 & 0.12 \end{bmatrix}.$$

B. Doubly stochastic matrix used in simulation experiment

$$\mathbf{H} = \begin{bmatrix} 0.57 & 0 & 0 & 0 & 0 & 0 & 0.20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.12 & 0 & 0.11 \\ 0 & 0.49 & 0 & 0 & 0 & 0.10 & 0.26 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.42 & 0 & 0.26 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.56 & 0 & 0 & 0 & 0.21 & 0 & 0 & 0 & 0 & 0 & 0.13 & 0 & 0.11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.26 & 0 & 0.61 & 0 & 0.06 & 0 & 0 & 0.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.10 & 0 & 0 & 0 & 0.40 & 0 & 0 & 0 & 0 & 0 & 0.28 & 0 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.20 & 0.26 & 0 & 0 & 0.06 & 0 & 0.48 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.21 & 0 & 0 & 0 & 0.52 & 0.27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0.27 & 0.39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.07 & 0 & 0 & 0 & 0 & 0.46 & 0 & 0 & 0 & 0 & 0 & 0.22 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 & 0 & 0 & 0 & 0.14 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.28 & 0 & 0 & 0 & 0 & 0 & 0.53 & 0.19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0.55 & 0 & 0 & 0.13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.07 & 0 & 0 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.55 & 0 & 0 & 0 & 0 & 0 & 0.16 \\ 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.68 & 0.21 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.22 & 0.14 & 0 & 0.13 & 0 & 0.21 & 0.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0.57 & 0 & 0.19 \\ 0.12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.53 & 0.18 & 0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.18 & 0.57 & 0 \\ 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.16 & 0 & 0 & 0.19 & 0.17 & 0 & 0.37 \end{bmatrix}.$$