Convex optimization based Sparse Learning over Networks

Ahmed Zaki and Saikat Chatterjee
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm
Emails: zakiah@kth.se, sach@kth.se

***Abstract*—In this paper, we consider the problem of estimating a sparse signal over a network. The main interest is to save communication resource for information exchange over the network and hence reduce processing time. With this aim, we develop a distributed learning algorithm where each node of the network uses a locally optimized convex optimization based algorithm. The nodes iteratively exchange their signal estimates over the network to refine the local estimates. The convex cost is constructed to promote sparsity as well as to include influence of estimates from the neighboring nodes. We provide a restricted isometry property (RIP)-based theoretical guarantee on the estimation quality of the proposed algorithm. Using simulations, we show that the algorithm provides competitive performance vis-a-vis a globally optimum distributed LASSO algorithm, both in convergence speed and estimation error.**

***Index Terms*—Sparse learning, convex optimization, greedy algorithms, restricted isometry property.**

I. Introduction

Learning of sparse signals/data from a limited number of observations has become important in many applications, for example, sparse coding, compressive sampling, dictionary learning [1], [2], [3], etc. In the gamut of sparse learning, use of convex optimization, mainly the ℓ_1-norm based minimization has been extensively studied [4]. Other class of sparse learning algorithms are greedy methods and Bayesian approaches [5]. The problem of sparse signal learning becomes more challenging over a distributed setup where observations are spread across nodes of a network. This distributed sparse learning problem is relevant in applications such as big data analysis [6], sensor networks [7], [8] etc. In such a scenario, the sparse signal learning involves learning and exchanging information among the nodes of the network. A straightforward approach would be to process the observations at a central node. This is expected to involve sending large amounts of data over the network, resulting in a high demand on communication resources. Furthermore, for security or privacy issues, observations and system parameters may not be accessible at a single place.

With this background, we design a distributed sparse learning algorithm referred to as network pursuit denoising (NBPDN) in this article. The algorithm solves a convex optimization problem to learn a local estimate. The estimates are exchanged between nodes of the network for further improvement. We expect the algorithm to converge in a limited number of iterations, thus saving communication resource and requiring a limited processing time. In this article, our main contributions are as follows:

- We develop an ℓ_1-norm minimization based distributed algorithm that achieves fast convergence. We construct the penalty function to include influence of estimates from the neighboring nodes.
- Using restricted-isometry-property based theoretical analysis, we state bounds on signal estimation quality.

A. Relation to Prior Work

We review relevant works on sparse learning over networks in this subsection. This problem has been attempted via greedy pursuit algorithms as well as the traditional ℓ_1 norm minimization based algorithms. The distributed greedy algorithms comprise of simple (and mostly heuristic) algorithmic steps and hence provide a computational advantage. In [9], a greedy algorithm is proposed that involves exchange of observations, estimates and observation matrix to reach a consensus on the estimation over the network. A distributed iterative hard thresholding algorithm is developed in [10] that provides sparse learning for both static and time-varying networks. Based on subspace pursuit [11] and CoSamp [12] algorithms used for centralized sparse learning, a set of distributed algorithms are proposed in [13] that provide a high computational advantage.

Use of ℓ_1-norm based convex optimization for sparse learning has been investigated with considerable interest due to its optimality and robust solutions. A distributed approach to solve the basis pursuit denoising (BPDN) [14] using the method of alternating-direction-method-of-multipliers (ADMM) was proposed in [15]. This algorithm referred to as distributed LASSO (D-LASSO), was shown to efficiently solve the distributed BPDN problem. Further work was done in [16] for the noiseless setting, that means for realizing distributed basis pursuit [14]. Works in [17], [18] have proposed distributed compressive sensing algorithms using convex optimization. At this point we mention that the D-LASSO [15] provides a globally optimum distributed solution using ADMM but suffers from slow convergence. The slow convergence of D-LASSO is shown in [9]. In contrast to D-LASSO, our interest is to develop convex optimization based algorithms that are fast in convergence, albeit at the expense of global optimality.

978-9-0827-9703-9/19/$31.00

B. System Model

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 $\mathbf{H} \in \mathbb{R}^{L\times L}$ where h_{lr} is the link weight from node r to node l. Our task is estimation of a sparse signal $\mathbf{x}$ in a distributed manner. The received signal (of size M_l) at node l can be written as

$$\mathbf{y}_l = \mathbf{A}_l\mathbf{x} + \mathbf{e}_l, \tag{1}$$

where $\mathbf{y}_l \in \mathbb{R}^{M_l}$, $\mathbf{A}_l \in \mathbb{R}^{M_l\times N}$ is the system matrix and $\mathbf{e}_l \in \mathbb{R}^{M_l}$ is an additive noise. We assume that $\mathbf{H}$ is a right stochastic matrix. This assumption is quite general in the sense that any non-negative network matrix can be recast as a right stochastic matrix by row normalization. Furthermore, the observation noise is assumed to be bounded, i.e., $\|\mathbf{e}_l\| \leq \epsilon$. This is a commonly used assumption in ℓ_1 minimization based sparse learning algorithms [19]. The signal of interest can be either exactly sparse ($\|\mathbf{x}\|_0 = s$) or approximately sparse (s highest amplitude elements of $\mathbf{x}$ contain the maximum energy of the signal). We use $\|\cdot\|_0$, $\|\cdot\|_1$ and $\|\cdot\|$ to denote the standard ℓ_0, ℓ_1 and ℓ_2 norm of an argument vector, respectively.

We use calligraphic letters $\mathcal{T}$ and $\mathcal{S}$ to denote sets that are sub-sets of $\Omega \triangleq \{1, 2, \ldots, N\}$. We use $|\mathcal{T}|$ and $\mathcal{T}^c$ to denote the 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 $(\cdot)^t$ and $(\cdot)^\dagger$ as transpose and pseudo-inverse, respectively. In this work $\mathbf{A}_\mathcal{T}^\dagger \triangleq (\mathbf{A}_\mathcal{T})^\dagger$. 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 support of a vector, as $\texttt{supp}(\mathbf{x}, s) \triangleq$ {the set of indices corresponding to the s largest amplitude components of $\mathbf{x}$}. If $\mathbf{x}$ has s non-zero elements then $\mathcal{T} = \texttt{supp}(\mathbf{x}, s)$. The *$s-$Restricted-Isometry-Constant* (RIC) [20] of a matrix $\mathbf{A}$ is denoted as δ_s. Additionally, the *$\{s, s'\}$-restricted orthogonality constant* (ROC) [21] of a matrix $\mathbf{A}$ is denoted as $\theta_{s,s'}$.

The rest of the paper is organized as follows. The NBPDN 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. Network Basis Pursuit Denoising

In this section, we describe the proposed network basis pursuit denoising (NBPDN) algorithm. The pseudo-code of the NBPDN is shown in Algorithm 1. As the name suggests, the structure of NBDN is motivated by the BPDN algorithm [14]. In the zeroth iteration, at each node we start with the standard BPDN where ϵ is used as an error bound (as $\|\mathbf{e}_l\| \leq \epsilon$). Then, for each iteration we solve a modified cost of BPDN where we add the penalty $g(\mathbf{x}, \{\hat{\mathbf{x}}_{r,k-1}, h_{lr}\}), r \in \mathcal{N}_l$. The penalty helps to incorporate the influence of estimates from all neighboring nodes.

Algorithm 1 NBPDN - Steps at Node l

Input: $\mathbf{y}_l$, $\mathbf{A}_l$, ϵ

Initialization:

$k \leftarrow 0$ (k denotes iteration counter)

$\hat{\mathbf{x}}_{l,0} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.t. $\|\mathbf{y}_l - \mathbf{A}_l\mathbf{x}\|_2 \leq \epsilon$ (BPDN)

Iteration:

repeat

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

$\hat{\mathbf{x}}_{l,k} = \arg\min_{\mathbf{x}} \lambda\|\mathbf{x}\|_1 + (1-\lambda) g(\mathbf{x}, \{\hat{\mathbf{x}}_{r,k-1}, h_{lr}\}), r \in \mathcal{N}_l$ s.t. $\|\mathbf{y}_l - \mathbf{A}_l\mathbf{x}\|_2 \leq \epsilon$

until *stopping criterion*

Output: $\hat{\mathbf{x}}_l$

For NBPDN, we define the $g(.)$ function as,

$$g(.) = \|\mathbf{x} - \sum_{r\in\mathcal{N}_l} h_{lr}\hat{\mathbf{x}}_{r,k-1}\|_1.$$

The use of $\sum_{r\in\mathcal{N}_l} h_{lr}\hat{\mathbf{x}}_{r,k-1}$ is a strategy for inclusion of past estimates from neighboring nodes. We use the additive strategy for simplicity and analytical tractability. If all the solutions of neighboring nodes are sparse then the additive term is also expected to be sparse. The use of ℓ_1 norm promotes sparsity on the difference signal $\left(\mathbf{x} - \sum_{r\in\mathcal{N}_l} h_{lr}\hat{\mathbf{x}}_{r,k-1}\right)$. Therefore, our hypothesis is that the ℓ_1 norm based $g(.)$ function promotes a sparse solution for $\mathbf{x}$. The solution is supposed to have a high overlap between its support and the support of the sparse signal $\sum_{r\in\mathcal{N}_l} h_{lr}\hat{\mathbf{x}}_{r,k-1}$. In the cost minimization, the parameter $\lambda \in [0, 1]$ needs to be carefully chosen to keep a balance between the sparsity promoting function $\|\mathbf{x}\|_1$ and the $g(.)$ function. Next, we mention our main theoretical result on bounding the estimation error.

Main Theoretical Result: For notational clarity, we use RIC constant $\delta_{s_1} \triangleq \max_l\{\delta_{s_1}(\mathbf{A}_l)\}$ and ROC constant $\theta_{s_1,s_2} \triangleq \max_l\{\theta_{s_1,s_2}(\mathbf{A}_l)\}$ where s_1, s_2 are constants. In the NBPDN algorithm, the estimation error at iteration k, $\mathbf{z}_{l,k} \triangleq \hat{\mathbf{x}}_{l,k} - \mathbf{x}$, follows a recurrence inequality, as stated in the following theorem.

Theorem 1 (Recurrence inequality): For NBPDN, under the condition $\delta_{s+a} + \sqrt{\frac{s}{b}}(2\lambda' - 1)\theta_{s+a,b} < 1$, at iteration k,

$$\|\mathbf{z}_{l,k}\|_1 \leq c_2 \sum_{r\in\mathcal{N}_l} h_{lr}\|\mathbf{z}_{r,k-1}\|_1 + c_3\|\mathbf{x}_{\mathcal{T}_0^c}\|_1 + c_4\epsilon,$$

where $\lambda' = \max\{\lambda, 1/2\}$, $c_2 = 2(1-\lambda)\left[1 + 2\lambda\sqrt{\frac{s}{b}}\frac{\theta_{s+a,b}}{c_5}\right]$, $c_3 = 2\lambda\left[1 + 2\lambda\sqrt{\frac{s}{b}}\frac{\theta_{s+a,b}}{c_5}\right]$, $c_4 = \frac{4\lambda\sqrt{s(1+\delta_{s+a})}}{c_5}$, and $c_5 = 1 - \delta_{s+a} - \sqrt{\frac{s}{b}}(2\lambda' - 1)\theta_{s+a,b}$.

In the above theorem, s is a parameter that determines the sparsity approximation of the signal $\mathbf{x}$. Also, $\mathcal{T}_0$ is the approximate sparsity index set of $\mathbf{x}$, i.e., $\mathcal{T}_0 = \texttt{supp}(\mathbf{x}, s)$.

We next state the result that under certain technical conditions and $c_1 = \delta_{s+a} + \sqrt{\frac{s}{b}}\theta_{s+a,b} < 1$, the NBPDN algorithm gives an estimate that is bounded. With the help of the recurrence inequality (Theorem 1), we state the following result for a general λ.

Theorem 2 (Bound on estimation error): If $c_1 = \delta_{s+a} + \sqrt{\frac{s}{b}}\theta_{s+a,b} < 1$, then at iteration k, the estimation error is bounded by

$$\|\mathbf{x} - \hat{\mathbf{x}}_{l,k}\| \leq d_{1k}\epsilon + d_{2k}\|\mathbf{x}_{\mathcal{T}_0^c}\|_1,$$

where $d_{1k} = \frac{c_2^k - 1}{c_2 - 1}(c_4 + c_2 c_7), d_{2k} = \frac{c_2^k - 1}{c_2 - 1}(c_3 + c_2 c_6), c_6 = 2\left[1 + 2\sqrt{\frac{s}{b}}\frac{\theta_{s+a,b}}{c_5}\right]$, and $c_7 = \frac{4\sqrt{s(1+\delta_{s+a})}}{c_5}$.
The proof of the above theorems is not shown here for brevity and will be discussed in an extended manuscript later.

The above bound holds at each node of the network. With $s = 4a = b$ and using the RIC and ROC properties [21], the condition $c_1 < 1$ reduces to $\delta_{2s} < 0.472$. Under the assumption of no observation noise ($\epsilon = 0$) and the signal being exactly s-sparse ($\mathbf{x}_{\mathcal{T}_0^c} = 0$), the NBPDN achieves exact estimate of $\mathbf{x}$ at every node when $\delta_{2s} < 0.472$. Note that the bound on the estimation error is a function of the iteration value k. This means that the bound increases with increase in iteration number, but remains finite when $\delta_{2s} < 0.472$.

A. Discussions

For the scenario of no cooperation over network, that is, if $\mathbf{H}$ is an identity matrix, NBPDN is same as the BPDN algorithm. Interestingly, it has been shown that the BPDN has an RIP condition that $\delta_{2s}(\mathbf{A}_l) < 0.472$ [21] for bounded reconstruction. Therefore, according to our analysis the RIP conditions for BPDN and NBPDN are comparable. It is interesting to compare NBPDN to other greedy pursuit based network algorithms such as network greedy pursuit (NGP) [22] and distributed hard thresholding pursuit (DHTP) [23] that solve a similar problem. The NGP and DHTP were shown to have RIP conditions of $\delta_{3s}(\mathbf{A}_l) < 0.362$ and $\delta_{3s}(\mathbf{A}_l) < 0.333$, respectively. It can be seen that RIP conditions for these two distributed greedy algorithms are more strict compared to the proposed NBPDN.

III. Simulation Results

In this section, we study the performance of the NBPDN algorithm using simulations. We first describe the simulation setup. Simulation results are then discussed.

A. Simulation Setup

We consider a randomly chosen connected network with L nodes where each node is connected to other d nodes. The parameter d referred to as the 'degree of the network' gives a measure of the network connection density. We have $d = |\mathcal{N}_l|$. Given the edge matrix $\mathcal{E}$ of the network, we can generate a right stochastic network matrix $\mathbf{H}$ by row normalization. For a fixed $\mathbf{H}$, we perform Monte-Carlo simulations where we randomly draw measurement matrices, Gaussian sparse signals and additive Gaussian noise. The performance metric used to compare the various algorithms is the mean signal-to-estimation-noise-ratio (mSENR), mSENR $= \frac{1}{L}\sum_{l=1}^{L}\frac{\mathbb{E}\{\|\mathbf{x}\|^2\}}{\mathbb{E}\{\|\mathbf{x}-\hat{\mathbf{x}}_l\|^2\}}$, where $\mathbb{E}(.)$ is the sampling average of the simulation. We define the signal-to-noise ratio (SNR) for node l as $\text{SNR}_l = \frac{\mathbb{E}\{\|\mathbf{x}\|^2\}}{\mathbb{E}\{\|\mathbf{e}_l\|^2\}}$. For simplicity, we assume that SNR is same at all the nodes, that is $\forall l, \text{SNR}_l = \text{SNR}$. Also, we consider the observation matrices to be of the same size, i.e., $\forall l, M_l = M$. We simulate D-LASSO and NBPDN algorithms. The reason to compare with D-LASSO is that it provides a benchmark performance – centralized solution of BPDN in a distributed manner. For the above algorithms the stopping criterion is assumed to be a corresponding maximum number of iterations allowed. This constraint allows us to enforce a measure of communication constraints and/or processing time constraints. For all the experiments, we set observation size $M = 100$ and signal dimension $N = 500$ with sparsity level $s = 20$. The network is assumed to have 20 nodes ($L = 20$) with degree, $d = 4$.

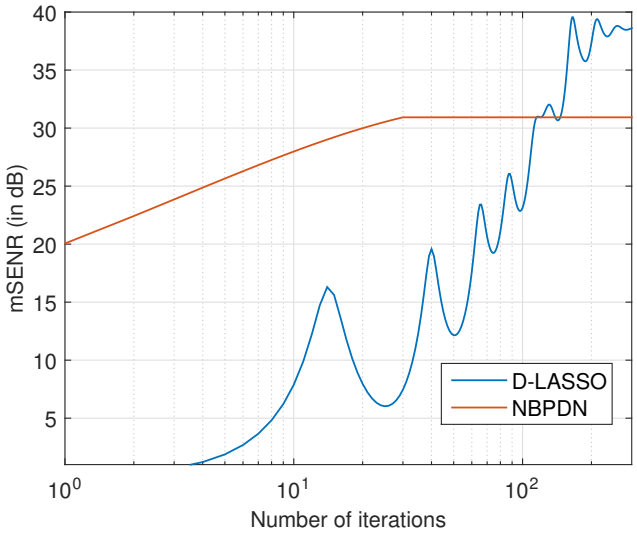

Fig. 1: Performance of various algorithms with respect to number of iterations over network (number of information exchanging iterations). We set $M = 100$, $N = 500$, $s = 20$, $L = 20$, $d = 4$, and SNR $= 30$dB. Performances are shown for 300 iterations where we use logarithmic scale to show iterations.

B. Experiment on Convergence Speed

In this experiment, we observe how fast the algorithms converge with iterations. We set SNR = 30dB and the maximum number of iterations be 300. In Fig. 1 , we show performance of the NBPDN algorithm and D-LASSO. We assumed that the network can support more communication such that D-LASSO can continue for 300 iterations. We see that D-LASSO has a

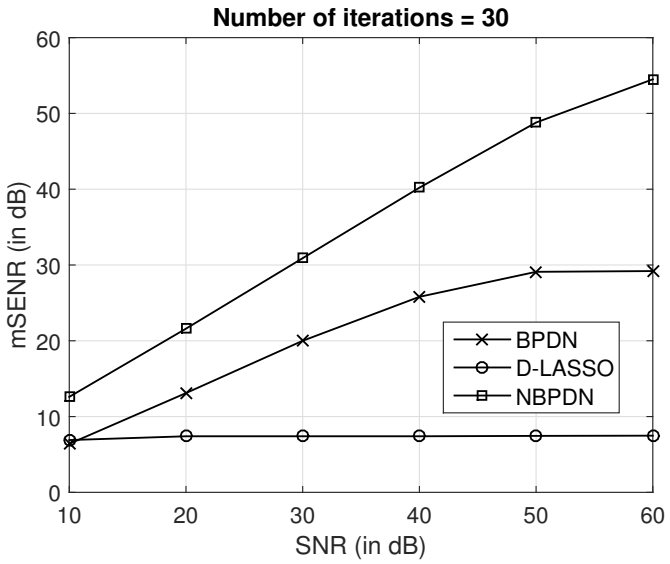

Fig. 2: Performance comparison of BPDN, D-LASSO and NBPDN algorithms with respect to SNR. We set M = 100, N = 500, s = 20, L = 20, d = 4 and maximum number of iterations = 30.

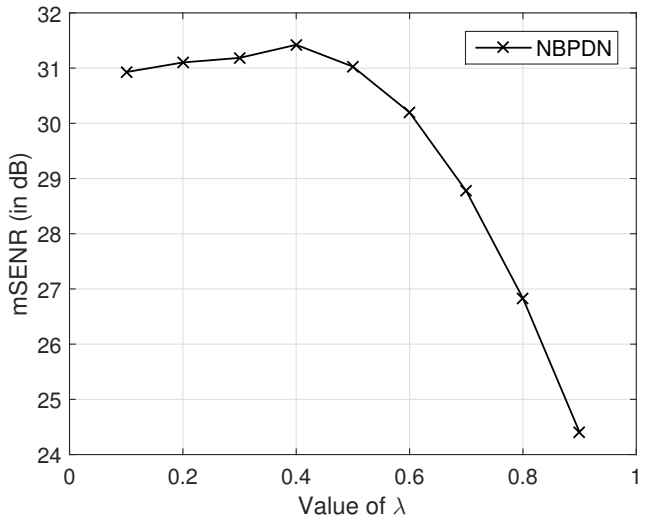

Fig. 3: Performance of algorithms with respect to the parameter λ. We set $M = 100$, $N = 500$, $s = 20$, $L = 20$, $d = 4$, and $SNR = 30\,dB$.

slow convergence, but provides best performance at the end of iterations. This is expected as it solves the centralized sparse learning problem using ADMM. It can be observed that the NBPDN converges around 30 iterations.

C. Experiment on Robustness to Measurement Noise

In this experiment, we investigate performance of the NBPDN algorithm at various SNR conditions to check robustness of algorithms to observation noise power. We show performance of D-LASSO, NBPDN and BPDN. BPDN does not cooperate over the network. Performances are shown in Fig. 2 where we set allowable numbers of iterations as 30. Note that D-LASSO hits a floor with increase in SNR. On the other hand NBPDN shows improving performance with the increase in SNR. It is interesting to note that for a communication constrained case and limited processing time, D-LASSO turns out to be even poorer than BPDN. On the other hand, the NBPDN shows good results for limited iterations.

D. Experiment on Sensitivity to Parameter λ

In all our previous experiments we have set the parameter $\lambda = 0.1$ for the proposed algorithms. This value was chosen based on an experiment where we vary λ for SNR = 30 dB and plot the performance in Fig. 3. It can be seen that the chosen λ is in the good performance region. A higher value of λ provides more weight to $\|\mathbf{x}\|_1$ and less weight to $g(\mathbf{x}, \{\hat{\mathbf{x}}_{r,k-1}, h_{lr}\})$. That means, more weight is assigned to sparsity promotion and less weight in using information from neighbors. It can be observed that good performance is achieved at lower value of λ where a higher importance is given to the information from neighboring nodes.

IV. Conclusion

We show that the NBPDN algorithm is good for a distributed learning setup where a sparse signal is estimated over a network. The NBPDN algorithm is fast in convergence, saving communication and computing resources. The algorithm is robust with additive noise model, and has a theoretical support on its performance. Our theoretical analysis results show that the restricted-isometry-property based estimation guarantees of the proposed NBPDN algorithm and the basis pursuit denoising (BPDN) algorithm is similar. The NBPDN algorithm can be used as a framework to design other convex algorithm by engineering appropriate regularization constraints for the $g(.)$ function.

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289 –1306, Apr. 2006.

[2] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*. Springer, 2010.

[3] 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, no. 2, pp. 349–396, Feb. 2003.

[4] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[5] W. Chen and I. J. Wassell, "A decentralized bayesian algorithm for distributed compressive sensing in networked sensing systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1282–1292, Feb. 2016.

[6] 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 Magazine*, vol. 31, no. 5, pp. 18–31, Sept 2014.

[7] T. Wimalajeewa and P. K. Varshney, "OMP based joint sparsity pattern recovery under communication constraints," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5059–5072, Oct. 2014.

[8] ——, "Wireless compressive sensing over fading channels with distributed sparse random projections," *IEEE Trans. Signal and Inf. Process. over Networks*, vol. 1, no. 1, pp. 33–44, Mar. 2015.

[9] 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.

[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] 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.

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

[13] 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, no. 11, pp. 2803–2818, June 2016.

[14] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

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

[16] 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.

[17] 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*, Nov. 2005, pp. 1537 – 1541.

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

[19] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math*, vol. 59, no. 8, pp. 1207–1223, 2006.

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

[21] T. T. Cai, L. Wang, and G. Xu, "Shifting inequality and recovery of sparse signals," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1300–1308, March 2010.

[22] 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.

[23] A. Zaki, P. P. Mitra, L. K. Rasmussen, and S. Chatterjee, "Estimate exchange over network is good for distributed hard thresholding pursuit," *Signal Processing*, vol. 156, pp. 1 – 11, 2019.