LEARNING DISTRIBUTED JOINTLY SPARSE SYSTEMS BY COLLABORATIVE LMS

Yuantao Gu[*] *Mengdi Wang*[†]

[*]State Key Laboratory on Microwave and Digital Communications
Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, CHINA
[†]Department of Operations Research and Financial Engineering
Princeton University, Princeton 08544, USA

ABSTRACT

In the proposed model of adaptive filtering network, distributed learning algorithm works cooperatively to identify separated unknown systems, which have different impulse responses. Specifically, JS-CoLMS algorithm is proposed to iteratively learn the unknown systems and the joint sparsity, based on a stochastic gradient approach and a subdifferentiable sparse-inducing penalty approximating the $l_{2,0}$ norm. The superior performance of the proposed algorithm and its relation to l_0-LMS and Leaky LMS are briefly discussed and verified by numerical experiments.

Index Terms— Distributed learning, adaptive filtering network, joint sparsity, Collaborative LMS, l_0-LMS, JS-CoLMS, Leaky LMS, $l_{2,0}$ norm, distributed optimization.

1. ADAPTIVE FILTERING NETWORK

We consider a network of M nodes, denoted as $\{N_m\}_{m=1}^M$, which are connected by directed edges. Supposing each node corresponds to an unknown system, the task is to learn "online" the coefficients of these systems utilizing their driven signals $\{x_m(n)\}_{m=1}^M$ and outputs $\{d_m(n)\}_{m=1}^M$, where n denotes the time instant. For this purpose, we will develop a distributed online learning algorithm, termed collaborative least mean squares (CoLMS). Assuming that each system can be represented by an L-order linear finite impulse response filter, one has

$$d_m(n) = \mathbf{h}_m^{\mathrm{T}}\mathbf{x}_m(n) + v_m(n), \quad \forall m = 1, 2, \cdots, M,$$

where

$$\mathbf{h}_m = [h_{0,m}, h_{1,m}, \cdots, h_{L-1,m}]^{\mathrm{T}},$$
$$\mathbf{x}_m(n) = [x_m(n), x_m(n-1), \cdots, x_m(n-L+1)]^{\mathrm{T}},$$

and $v_m(n)$ denote, respectively, the unknown system response, the training vector, and the additive measurement noise. For online estimation of the system coefficients $\{\mathbf{h}_m\}_{m=1}^M$, gradient descent-based adaptive filtering algorithms are quite competitive for their low computational complexity, robustness, and easy implementation.

In the absence of any prior knowledge regarding these unknown systems, this problem may fall into the traditional study of adaptive filter, which has been thoroughly investigated in numerous literatures. However, when there exists certain prior information, we may adapt the filters in a collaborative manner to improve the overall performance. In this paper, we will focus on an important case of *joint sparsity*, meaning that the support set of each system response is identical.

As far as we know, using adaptive filtering network to learn distributed and heterogeneous systems, based on joint sparsity or other priors, has not been considered in literatures. However, under the assumption that the unknown systems are exactly identical, i.e., $\mathbf{h}_m \equiv \mathbf{h}_1$ for all m, there were comprehensive and solid works, including incremental LMS [1, 2], incremental RLS [1], diffusion RLS [3, 4], and diffusion LMS [1, 5]. The later has been exhaustively studied from various aspects, including information exchange methods [6], fundamental limits [7], and sparse constraint [8]. It should be noticed that joint sparsity has been imposed and studied in the area of Distributed Compressive Sensing [9, 10], array signal processing [11], and wideband ADC [12]. Their motivations purport that the distributed learning under the joint sparsity assumption is an important subject of study.

In the next section, we will formulate the mentioned learning task into a centralized optimization problem, and will propose a new adaptive algorithm for its solution. In section 3, we will show that the proposed algorithm can be implemented in a distributed manner over networked adaptive filters.

2. COLLABORATIVE LMS EXPLORING JOINT SPARSITY (JS-COLMS)

Recall that our purpose is to estimate the responses of the M unknown systems with the tap-weights of M adaptive filters, $\mathbf{w}_m = [w_{0,m}, w_{1,m}, \cdots, w_{L-1,m}]^{\mathrm{T}}, m = 1, 2, \cdots, M$, which can be combined as

$$\mathbf{W} = \begin{bmatrix} w_{0,1} & w_{0,2} & \cdots & w_{0,M} \\ w_{1,1} & w_{1,2} & \cdots & w_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ w_{L-1,1} & w_{L-1,2} & \cdots & w_{L-1,M} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{w}}_0^{\mathrm{T}} \\ \bar{\mathbf{w}}_1^{\mathrm{T}} \\ \vdots \\ \bar{\mathbf{w}}_{L-1}^{\mathrm{T}} \end{bmatrix},$$

where $\bar{\mathbf{w}}_k = [w_{k,1}, w_{k,2}, \cdots, w_{k,M}]^{\mathrm{T}}, k = 0, 1, \cdots, L-1$, is a vector composed of the kth tap-weights of all the M filters.

Let us begin by taking a centralized optimization viewpoint. Under the joint sparsity assumption, one may expect that zeros dominate the set of $\{\|\bar{\mathbf{w}}_k\|_2\}_{k=0}^{L-1}$, where the l_2 norm is used to merge the coefficients at the same position of all filters into a single quantity. As a result, the level of joint sparsity of the filters can be character-

[*]This work was partially supported by the National Program on Key Basic Research Project (973 Program 2013CB329201) and the National Natural Science Foundation of China (NSFC 61371137). The corresponding author of this paper is Yuantao Gu (gyt@tsinghua.edu.cn).

ized by the $l_{2,0}$ norm of $\mathbf{W}$, which is defined as

$$\|\mathbf{W}\|_{2,0} = \left\| \left[\|\bar{\mathbf{w}}_0\|_2, \|\bar{\mathbf{w}}_1\|_2, \cdots, \|\bar{\mathbf{w}}_{L-1}\|_2 \right]^{\mathrm{T}} \right\|_0$$

where $\|\cdot\|_0$ denotes the number of nonzero entries of a given vector. We formulate the learning problem of jointly sparse systems as a regularized optimization problem,

$$\min_{\mathbf{W}} \left(\xi(\mathbf{W}) = \sum_{m=1}^{M} \mathrm{E}\left\{ |e_m|^2 \right\} + \gamma \|\mathbf{W}\|_{2,0} \right), \tag{1}$$

where

$$e_m = d_m - \mathbf{w}_m^{\mathrm{T}} \mathbf{x}_m \tag{2}$$

is the estimation error at node N_m, the expectation $\mathrm{E}\{\cdot\}$ is taken over the steady-state distribution of the process $\{\mathbf{x}_m(n)\}_n$, and γ is a positive scalar to balance the sparsity penalty and the estimation error.

We attempt to solve problem (1) using a gradient descent method of the form

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \frac{\mu}{2} \nabla \xi(\mathbf{W}(n)), \tag{3}$$

where $\mu > 0$ denotes the stepsize. However, this does not work for two reasons: (i) the $l_{2,0}$ norm is discontinuous so that its gradient does not exist; and (ii) the expectation in (1) is difficult to compute. To make this approach practicable, we approximate the $l_{2,0}$ norm by using a subdifferentiable function as

$$\|\mathbf{W}\|_{2,0} \approx J(\mathbf{W}) = \sum_{k=0}^{L-1} F\left(\|\bar{\mathbf{w}}_k\|_2 \right), \tag{4}$$

where we choose a kind of F (see [13]) such that its subgradient takes the form

$$f_\alpha(t) = \begin{cases} \sqrt{M}\alpha \left(1 - \alpha t / \sqrt{M}\right) & 0 < t < \sqrt{M}/\alpha; \\ 0 & \text{elsewhere}, \end{cases} \tag{5}$$

and α is a positive parameter to control the accuracy of approximation. Note from (5) that the active region and strength of $f_\alpha(\cdot)$ is zoomed in $\sqrt{M}$ times, compared to [13], to counteract the impact of network size on parameter α. Consequently we approximate the subgradient of the approximate $l_{2,0}$ norm of (4) by

$$\frac{\partial J(\mathbf{W})}{\partial w_{k,m}} \approx \frac{f_\alpha\left(\|\bar{\mathbf{w}}_k\|_2 \right) w_{k,m}}{\|\bar{\mathbf{w}}_k\|_2 + \delta}, \tag{6}$$

where δ is a small positive quantity to avoid dividing zero. To deal with the second difficulty, we apply the stochastic gradient idea, i.e., to replace the expectation $\mathrm{E}\{|e_m|^2\}$ in (1) with its transient sample $|e_m(n)|^2$ at time n. By using (6) and the sample gradient, one may modify (3) to derive the learning recursion of the kth tap-weight at node N_m as

$$\begin{aligned} w_{k,m}(n+1) = & \left(1 - \frac{\kappa f_\alpha\left(\|\bar{\mathbf{w}}_k(n)\|_2 \right)}{\|\bar{\mathbf{w}}_k(n)\|_2 + \delta}\right) w_{k,m}(n) \\ & + \mu e_m(n) x_m(n-k), \end{aligned} \tag{7}$$

where $\kappa = \mu\gamma/2$ is the stepsize for zero-point attraction [13]. The detailed JS-CoLMS algorithm is given in TABLE 1.

Table 1. The Procedure of JS-CoLMS.

Input: $\{\mathbf{x}_m(n), d_m(n)\}_{m=1}^{M}, \mu, \kappa$;
Initialization: $\{\mathbf{w}_m(0) = \mathbf{0}\}_{m=1}^{M}, \delta = 1\mathrm{E}{-}10$;
Output: $\{\mathbf{w}_m(n)\}_{m=1}^{M}$.

For $n = 0, 1, 2, \cdots$
1) Filtering and estimation:
Calculate $\{e_m(n)\}_{m=1}^{M}$ by (2);
2) Adaptation:
Calculate $\{\mathbf{w}_m(n+1)\}_{m=1}^{M}$ by (7);
End

3. DISTRIBUTED IMPLEMENTATION OF JS-COLMS

JS-CoLMS in TABLE 1 works in a centralized way. However, it could be readily modified to a distributed implementation. According to (7), the adaptation of each tap-weight consists of two steps, namely the stochastic gradient descent and the zero-point attraction. Taking the example of node N_m, the gradient descent could be performed first to utilize the newly acquired training data and yield a temporary tap-weight vector $\mathbf{u}_m(n)$, which will then be communicated to the neighboring node for collaboration. Simultaneously, N_m will also receive the temporary tap-weights from its collaborators. After the information exchange process, N_m could calculate the zero-point attraction by utilizing the collaborative information. Finally the tap-weight $\mathbf{w}_m(n+1)$ is prepared for the next iteration. Distributed implementation of JS-CoLMS is described in detail in TABLE 2.

4. DISCUSSIONS

We will briefly discuss the performance gain of the proposed algorithm and its relation to existing reference algorithms. Further theoretical analysis is beyond the scope of the current paper.

JS-CoLMS will degenerate to standard l_0-LMS [13] when there is no collaboration,

$$w_{k,m}(n+1) = w_{k,m}(n) + \mu e_m(n) x_m(n-k) - \kappa f_\alpha(w_{k,m}(n)). \tag{8}$$

The faster convergence of (8) as compared to LMS is due to the zero-point attraction $\kappa f_\alpha(w_{k,m}(n))$, which additionally drags small coefficients to null. By the nature of LMS, the transient tap-weight contains strong stochastic gradient noise, especially before the steady state is arrived. Compared to l_0-LMS, the proposed JS-CoLMS synthesizes collaborators' information, $\|\bar{\mathbf{w}}_k(n)\|_2$, to yield a better attracting performance by exploiting the joint sparsity. Therefore, the latter could produce faster convergence rate and reach smaller steady-state deviation with the same control parameters as the former. It could be further speculated that as the number of collaborators increases, the performance improvement will monotonically approach an upper bound.

As to its distributed implementation, when there is no collaboration, JS-CoLMS degenerates to a form of l_0-LMS where the gradient descent and zero-point attraction are performed in two successive steps rather than simultaneously.

One important reason that we formulate JS-CoLMS in a one-step recursion of (7) is to hint its relation with a variable Leaky

Table 2. Distributed Implementation of JS-CoLMS at Node N_m.

Input: $\mathbf{x}_m(n), d_m(n), \mu, \kappa$,
$\mathcal{C}_m = \{m' | N_{m'}$ consumes information from $N_m\}$,
$\mathcal{P}_m = \{m\} \cup \{m' | N_{m'}$ provides information to $N_m\}$;
Initialization: $\mathbf{w}_m(0) = \mathbf{0}, \delta = 1\mathrm{E}-10$;
Output: $\mathbf{w}_m(n)$.

For $n = 0, 1, 2, \cdots$
1) Filtering and estimation:
$e_m(n) = d_m(n) - \mathbf{w}_m^{\mathrm{T}}(n)\mathbf{x}_m(n)$;
2) Adaptation of gradient descent:
$\mathbf{u}_m(n) = \mathbf{w}_m(n) + \mu e_m(n)\mathbf{x}_m(n)$;
3) Information exchange with collaborators:
Send $\mathbf{u}_m(n)$ to $N_{m'}, \forall m' \in \mathcal{C}_m$;
Receive $\mathbf{u}_{m'}(n)$ from $N_{m'}, \forall m' \in \mathcal{P}_m \setminus \{m\}$;
4) Adaptation of zero-point attraction:
For $k = 0, 1, \cdots, L-1$

$$t = \left(\sum_{m' \in \mathcal{P}_m} |u_{k,m'}(n)|^2 \right)^{1/2};$$

$$f_\alpha = \begin{cases} \sqrt{|\mathcal{P}_m|}\alpha\left(1 - \alpha t / \sqrt{|\mathcal{P}_m|}\right) & 0 < t < \sqrt{|\mathcal{P}_m|}/\alpha; \\ 0 & \text{elsewhere;} \end{cases}$$

$$w_{k,m}(n+1) = \left(1 - \frac{\kappa f_\alpha}{t + \delta}\right) u_{k,m}(n);$$

End
End

LMS[14]. Actually, (7) readily reminds us

$$w_k(n+1) = (1 - \mu\beta(n))w_k(n) + \mu e(n)x(n-k),$$

where $\beta(n) = f_\alpha\left(\|\bar{\mathbf{w}}_k(n)\|_2\right) / \left(\|\bar{\mathbf{w}}_k(n)\|_2 + \delta\right)$ denotes a time-varying leaky parameter. Therefore, our previous discussion on the robustness of JS-CoLMS is consistent with the feature of variable Leaky LMS, i.e., to *improve stability in a finite-precision implementation, $\cdots$, and reduce undesirable effects like stalling, bursting, etc*[14].

5. NUMERICAL SIMULATION

In this section, the proposed Collaborative LMS is tested in several scenarios of jointly-sparse-system identification. [1] For comparison, the algorithms to be simulated include traditional LMS, l_0-LMS[13], centralized and distributed JS-CoLMS.

The first experiment demonstrates the performance gain of JS-CoLMS from utilizing joint sparsity, based on a fully connected 6-node network. JS-CoLMS works on all nodes collaboratively, where the reference algorithms work individually on each node. The distributed unknown systems are driven by independent white Gaussian signals. Uncorrelated additive white Gaussian noises are available on all system outputs with an identical signal-to-noise ratio of 20dB.

[1] The code for these experiments is available at `http://gu.ee.tsinghua.edu.cn/publications#gu1`

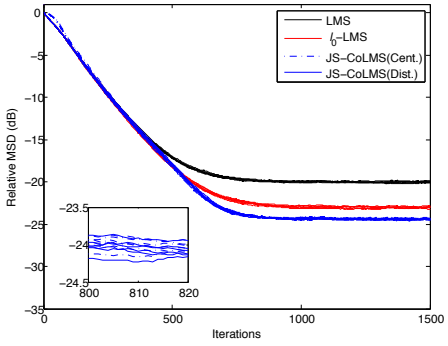

Fig. 1. The learning curves of various algorithms in the first experiment, where the network is composed of 6 fully connected nodes, whose corresponding unknown systems are with joint sparsity of $64/128$.

The six 128-order system responses share the same nonzero coefficients support set, which is randomly chosen among all possible choices and has a cardinality of 64; each nonzero coefficient is randomly generated by standard Gaussian distribution. The algorithm parameters are set as: $\mu = 1/128$, $\alpha = 10$, and $\kappa = 2\mathrm{E}-4$. The relative mean squared deviation from 100 independent learning trails is calculated and plotted in Fig. 1. Note that for each algorithm there is a bundle of 6 learning curves corresponding to the 6 nodes. As can be seen from Fig. 1, joint sparsity is helpful in reducing the steady-state error while maintaining fast convergence rate. Also note that the performances of distributed and centralized JS-CoLMS are quite close to each other, so in what follows we only experiment with the distributed version.

The second experiment studies the relation between the number of collaborators and the performance gain of collaborative LMS. The setup and all algorithms' parameters are the same as the first experiment, except that we vary the network size M from 1 to 10. Then the statistics, including mean and deviation, of relative squared deviation are plotted with respect to M in Fig. 2. According to Fig. 2, collaborators more than 4 improve little performance. On the other hand, one may be happy to recognize that a few collaborators are sufficient to produce a rather good steady-state performance.

Our third experiment studies how the level of joint sparsity affects the performance of JS-CoLMS. We use the same simulation settings and parameters as in the first experiment, except that we vary the number of nonzero coefficients to 8, 16, 32, 64, and 128. The steady-state statistics of the tested algorithms with respect to different sparsity levels are plotted in Fig. 3. As has been conjectured, JS-CoLMS yields consistently better performance than l_0-LMS, especially when the number of nonzero entries is small.

Our last experiment is to imitate a *realistic* scenario. Suppose in a unit square area there are 1000 nodes with coordinates $\{(i_m, j_m)\}_{m=1}^{1000}, i_m, j_m \in (0, 1)$. The kth unknown coefficient corresponding to N_m is generated by

$$h_{k,m} \begin{cases} \sim \mathcal{N}(0,1), & k = 0, 1, \cdots, 31 + 96 i_m; \\ = 0, & \text{elsewhere.} \end{cases}$$

The collaborative relation between N_m and $N_{m'}$ is determined by

$$\left.\begin{matrix} m' \in \mathcal{P}_m \\ m \in \mathcal{C}_{m'} \end{matrix}\right\} \text{if} \left\{\begin{matrix} \text{dist}(m, m') \le 0.063\sqrt{j_m} \\ \text{and} \quad i_{m'} \le i_m, \end{matrix}\right.$$

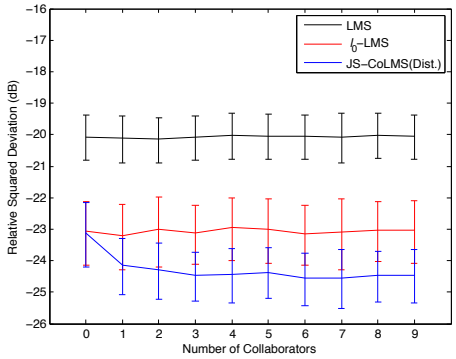

Fig. 2. The statistics of the steady-state square deviation of various algorithms in the second experiment, with respect to the number of collaborators, while the unknown joint sparsity is $64/128$.

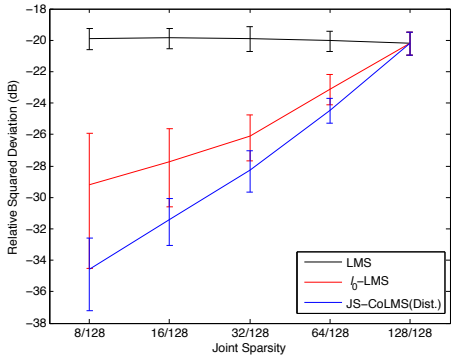

Fig. 3. The steady-state statistics in the third experiment, with respect to different joint sparsity level for a 6-node fully connected network.

where $\text{dist}(m, m') = \sqrt{(i_m - i_{m'})^2 + (j_m - j_{m'})^2}$ denotes the distance between N_m and $N_{m'}$. One may refer to Fig. 4, where circle-symbols denote the node and lines denote the directed collaborations from left to right. The diameter of circle denotes the number of nonzero coefficients. After 1000 trials, the steady-state MSD gains from exploring joint sparsity and their histogram are plotted in Fig. 5, where blue denotes positive gain and red negative, and Fig. 6, respectively. The results are self-evident that dominant number of nodes obtain joint sparsity gain and a few collaborators are sufficient to provide a visible improvement on accuracy, which is further enhanced by the decrease of joint number of nonzero coefficients.

6. CONCLUSION

This paper proposes a framework of adaptive filtering network, which can be considered as a generalization of distributed learning from a single system to noisy networked systems. Under this framework, it proposes an algorithmic realization of constrained LMS to explore the joint sparsity over a network of separate but related systems. Numerical experiments have been given to verify the effectiveness of this distributed algorithm.

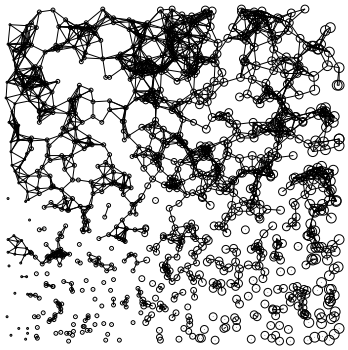

Fig. 4. The collaborative relation and the number of nonzero coefficients of respective node (circle) in the fourth experiment, where the number of nonzero coefficients (the diameter of circle) increases from left side to right side and the number of collaborators (the number of connected nodes at one's left side) increases from bottom side to top side.

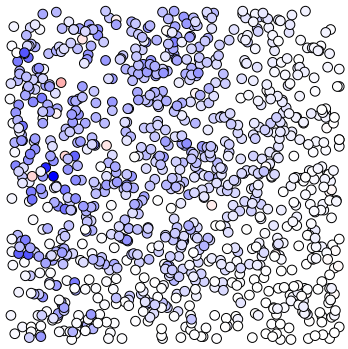

Fig. 5. The gain of learning accuracy, i.e, JS-CoLMS exceeds l_0-LMS, of respective nodes in the fourth experiment, where blue and red denote positive and negative gain, respectively.

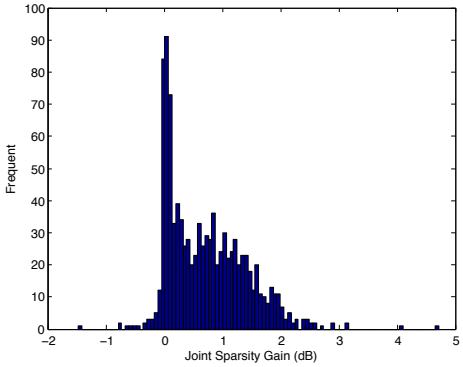

Fig. 6. The histogram of learning accuracy gain in the fourth experiment.

7. REFERENCES

[1] A. H. Sayed and C. G. Lopes, Adaptive processing over distributed networks, *IEICE Trans. Fund. Electron., Commun. Comput. Sci.*, E90-A(8):1504-1510, August 2007.

[2] C. G. Lopes and A. H. Sayed, Incremental adaptive strategies over distributed networks, *IEEE Trans. Signal Process.*, 55(8): 4064-4077, August 2007.

[3] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, A diffusion RLS scheme for distributed estimation over adaptive networks, *Proc. IEEE SPAWC*, Helsinki, Finland, June 2007, 1-5.

[4] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, *IEEE Trans. Signal Process.*, 56(5): 1865-1877, May 2008.

[5] C. G. Lopes and A. H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, *IEEE Trans. Signal Process.*, 56(7): 3122-3136, July 2008.

[6] F. S. Cattivelli and A. H. Sayed, Diffusion LMS Strategies for Distributed Estimation, *IEEE Trans. Signal Processing*, 58(3):1035-1048, March 2010.

[7] J. M. F. Moura and S. Kar, Convergence Rate Analysis of Distributed Gossip (Linear Parameter) Estimation: Fundamental Limits and Tradeoffs, *arXiv:1011.1677v1 [cs.IT]*, 7 November 2010.

[8] A. H. Sayed and P. Di Lorenzo, Sparse Distributed Learning Based on Diffusion Adaptation, *IEEE Trans. Signal Processing*, 61(6):1419-1433, June 2013.

[9] M. F. Duarte, S. Sarvotham, M. B. Wakin, D. Baron, and R. G. Baraniuk, Joint Sparsity Models for Distributed Compressed Sensing, http://spars05.irisa.fr/ACTES/TS5-2.pdf

[10] M. B. Wakin, M. F. Duarte, S. Sarvotham, D. Baron, and R. G. Baraniuk, Recovery of Jointly Sparse Signals from Few Random Projections, *Proc. Neural Information Processing Systems* (NIPS), December 2005.

[11] P. T. Boufounosa, P. Smaragdisb, and B. Raj, Joint Sparsity Models for Wideband Array Processing, *Proc. SPIE, Wavelets and Sparsity XIV*, September 27, 2011.

[12] M. Mishali and Y. C. Eldar, From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals, *IEEE Journal of Selected Topics in Signal Processing*, 4(2):375-391, April 2010.

[13] Y. Gu, J. Jin, and S. Mei, l_0 Norm Constraint LMS Algorithm for Sparse System Identification, *IEEE Signal Processing Letters*, 16(9):774-777, 2009.

[14] M. Kamenetsky and B. Widrow, A Variable Leaky LMS Adaptive Algorithm, *the 38th Asilomar Conference on Signals, Systems and Computers*, November 2004.