Distributed Spectrum Sensing in Cognitive Radios via Graphical Models

Alireza Makhzani
Department of Electrical and Computer Engineering
University of Toronto
makhzani@comm.utoronto.ca

Shahrokh Valaee
Department of Electrical and Computer Engineering
University of Toronto
valaee@comm.utoronto.ca

***Abstract*—There is a recent interest in developing algorithms for the reconstruction of jointly sparse signals, which arises in a large number of applications. In this work, we study the problem of wide-band spectrum sensing for cognitive radio networks using compressed sensing to exploit the underlying joint sparsity structure in a distributed setting. In particular, we use the recently proposed Approximate Message Passing (AMP) framework and exploit the spatial correlation that exists locally between different CRs in a non-centralized fashion. We will show that with the suggested scheme, the nodes iteratively exploit the local spatial information and achieve the consensus on the spectrum in a distributed fashion.**

***Index Terms*—Cognitive Radio, Distributed Spectrum Sensing, Compressed Sensing, Belief Propagation**

I. Introduction

The goal of Cognitive Radio (CR) is to increase the utilization of the underutilized frequency bands. Spectrum sensing [1] plays an important role in cognitive radio since secondary users need to detect primary signals in order to make decisions about the occupancy of the spectrum bands.

Compressive sensing theory has been previously considered in wideband spectrum sensing techniques [2]. In [3], a distributed compressive spectrum sensing scheme has been proposed which uses SOMP algorithm [4] to recover the jointly sparse signals (JSM) [5]. Other algorithms such as MUSIC [6], ReMBo [7] have been proposed in the literature for reconstructing such signals under this JSM model. In [8], the authors use an approximate message passing (AMP) [9], [10] framework in a factor graph to incorporate the spatial information. The signals are all collected at the fusion center and all the signals are reconstructed jointly in a synchronized and centralized fashion. After reconstructions, the signals are transmitted back to the individual nodes. But this method, can not be used in an ad-hoc network where there is no fusion center and a node might not see all the other nodes. In the ad-hoc networks there is still a considerable amount of spatial information among neighbors that could be exploited in a distributed fashion to reduce the number of measurements in compressed sensing.

In the present paper, we consider a network of distributed CRs. Different CRs receive the same wide-band signal at different SNRs. Each signal, experiences different attenuation and multi-path effect, which cause different amplitudes and phases, but we can expect that the locations of the excited frequency bands to be roughly the same. We use the Joint Sparsity Model (JSM) proposed in [5] to model the underlying structure of the signals. We propose a distributed and non-centralized algorithm that could be adopted by ad-hoc networks to incorporate the spatial information into compressed sensing. Each individual CR, reconstructs its own signal separately in a distributed fashion as follows. After each iteration, each CR wakes up and transmits its current estimate of the likelihood probability of the spectrum to all its neighbors and receives the likelihood probability of all its neighbors. Each CR then uses this information to update their estimate of the spectrum and the process is repeated again until a consensus on the spectrum is achieved. We call this algorithm "Distributed Compressed Sensing AMP (DCS-AMP)" and show that the DCS-AMP algorithm has a better performance than the state of the art algorithms, such as DCS-SOMP [5] and S-IHT [11].

II. Problem Formulation

Suppose that a total spectrum of W Hz is considered to be shared among a number of primary and secondary users in an ad-hoc network and $N = \frac{W}{B}$ is the number of available channels. Assume we have S CR sensing receivers. Each node is provided with a different $M \times N$ random measurement matrix Φ_s whose columns are normalized to have a unit ℓ_2 norm. Each CR receiver collects compressive time-domain samples using the matrix Φ_s. The received signals are sparse in the Fourier domain because of the sparsity in the occupancy of frequency bands. However, due to the universality of compressive sensing, without loss of generality, we can assume that the sparsity basis matrix is the identity matrix. Hence, we assume the signals are in the sparse domain and $x_{i,s}$ denotes the frequency content of the i-th channel of the received signal at the $s-$th receiver. So the compressive samples are obtained as follows: $\mathbf{y}_s = \Phi_s \mathbf{x}_s$. The sampling scheme to obtain the vector $\mathbf{y}_s$ is based on using an analog-to-information-converter (AIC) [12].

Let $x^t_{i,s}$ denote the $i-$th coefficient of the estimation of vector $\mathbf{x}_s$ at the $t-$th iteration. We model the spatial correlation by assuming that the received signals have exactly the same support set across all sensors but maybe with different coefficients. We further assume that $\mathbf{x}_{i,s}$ is coming from the following prior distribution

$$x_{i,s} \sim H(x_{i,s}) = p_{i,s}\mathcal{N}(0, \sigma_0^2) + (1 - p_{i,s})\mathcal{N}(0, \sigma_1^2)$$

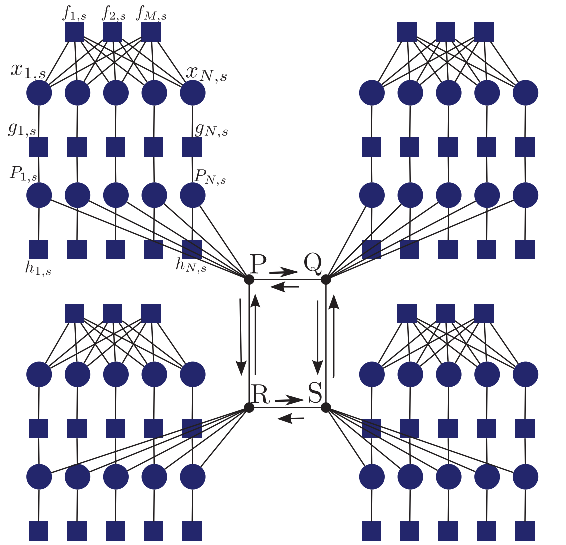


Fig. 1. Factor graph of DCS-AMP algorithm and the connectivity graph of the network

where $\sigma_1 \gg \sigma_0$ and $p_{i,s}$ is close to one for each coefficient of the shared support set. We model the noise by assuming a non-zero variance σ_0^2 for the coefficients that are not in the support set. We have a distinct graph in each node, with inputs arriving from the neighbors of the node in the network. For example, in Fig. 1, node P can transmit and receive just from node Q and node R. After each iteration, each sensor wakes up and transmits a message about the estimation of its own support set to all the other neighbors and receives all the messages from the neighbors. As all these messages arrive at each node, the internal graph of the corresponding node updates its values and a new marginal distribution for $\mathbf{x}_s$ is calculated separately in a distributed fashion. We will prove that in the long run, a state of agreement on the support set is achieved and the performance is just slightly worse than the centralized case where all the sensors have access to all the estimates at all the iterations through the fusion center. Throughout this paper, by $F \propto G$ we mean the function $F = \frac{1}{Z}G$, where Z is a normalization factor that makes G a valid density function.

Consider the following joint distribution on the coefficients of the signal.

$$f(\mathbf{x}) = \frac{1}{Z}\prod_{s=1}^{S}\prod_{i=1}^{N} H(x_{i,s}) \prod_{s=1}^{S}\prod_{j=1}^{M} \delta_{y_{j,s}=(\Phi_s \mathbf{x}_s)_j}$$

where $\mathbf{x}$ contains the coefficients of all signals and $H(x_{i,s}) = p_{i,s}\mathcal{N}(0,\sigma_0^2) + (1-p_{i,s})\mathcal{N}(0,\sigma_1^2)$ is the Gaussian mixture prior. We consider the factor graph of Fig. 1. This factor graph contains two types of variable nodes. The first type of variable nodes is $x_{1,s}, x_{2,s}, ..., x_{N,s}$ representing the coefficients of the signal and we call them the "coefficient nodes". Also, associated with each coefficient $x_{i,s}$ of the signal, there exists a hidden random variable $P_{i,s} \in \{0,1\}$ defining the state of the coefficients and we call them the "state nodes". If $P_{i,s} = 0$, then the coefficient comes from the small-variance Gaussian distribution and if $P_{i,s} = 1$, the coefficient comes from the large-variance Gaussian distribution. The factor graph also contains three types of function nodes. Each $f_{j,s}$ is a delta function on the hyperplane $y_{j,s} = (\Phi_s \mathbf{x}_s)_j$. The function $g_{i,s}$ is the conditional density function and $h_{i,s}$ accounts for the prior probability of $x_{i,s}$ being in the support set, which is defined by $q_{i,s}$. For the single sensor case, $q_{i,s} = p_{i,s}$. But for the multiple sensor case, as we will discuss, the spatial correlation contributes to $p_{i,s}$. The sum-product belief propagation algorithm for this factor graph is as follows.

$$\mu_{x_{i,s}\to f_{j,s}}(x_{i,s}) \propto H(x_{i,s}) \prod_{k\neq j} \mu_{f_{k,s}\to x_{i,s}}(x_{i,s})$$

$$\mu_{f_{j,s}\to x_{i,s}}(x_{i,s}) \propto \int \delta(y_{j,s}-(\Phi_s \mathbf{x}_s)_j) \prod_{k\neq i} \mu_{x_{k,s}\to f_{j,s}}(x_{k,s})\, d\mathbf{x}_s^{-i}$$

where $\mathbf{x}_s^{-i}$ includes all the coefficients of x_s except $x_{i,s}$.

$$\mu_{g_{i,s}\to x_{i,s}}(x_{i,s}) \propto H(x_{i,s}) = p_{i,s}\mathcal{N}(0,\sigma_0^2) + (1-p_{i,s})\mathcal{N}(0,\sigma_1^2)$$

$$\mu_{x_{i,s}\to g_{i,s}}(x_{i,s}) \propto \prod_k \mu_{f_{k,s}\to x_{i,s}}(x_{i,s})$$

$$\mu_{g_{i,s}\to P_{i,s}}(P_{i,s}) = \begin{bmatrix} p'_{i,s} \\ 1-p'_{i,s} \end{bmatrix} \propto \int \Pr(x_{i,s}|P_{i,s}) \prod_k \mu_{f_{k,s}\to x_{i,s}}(x_{i,s}) dx_{i,s}$$

$$\mu_{h_{i,s}\to P_{i,s}}(P_{i,s}) \propto h(P_{i,s}) = \begin{bmatrix} q_{i,s} \\ 1-q_{i,s} \end{bmatrix}$$

After each iteration, the nodes wake up and transmit the likelihood probability of the support set to all their neighbors which is denoted by $N(s)$.

$$\begin{aligned} \mu_{P_{i,s}\to g_{i,s}} &= \begin{bmatrix} p_{i,s} \\ 1-p_{i,s} \end{bmatrix} \propto h_{i,s} \prod_{k\in N(s)} \mu_{g_{i,k}\to P_{i,s}} \\ &= \begin{bmatrix} q_{i,s} \\ 1-q_{i,s} \end{bmatrix} \prod_{k\in N(s)} \begin{bmatrix} p'_{i,k} \\ 1-p'_{i,k} \end{bmatrix} \end{aligned}$$

Thus, essentially two sources of additional information are contributing to $p_{i,s}$. The first information is $q_{i,s}$ which comes from the prior information about the support set of the signals and will not change during the algorithm. The second information comes from the likelihood of the neighbor nodes. This information contributes to $p_{i,s}$ according to the equation of $\mu_{P_{i,s}\to g_{i,s}}$.

At the end of the $t-$th iteration, each sensor node has a belief about the probability of each coefficient of its associated signal being in the support set. These beliefs are of the form of likelihood functions and are being sent through the conditional function nodes $g_{i,s}$ to the state nodes $P_{i,s}$. We can prove that these likelihood functions can be approximated in the following Gaussian form using the AMP framework discussed in the next section.

$$\mathcal{L}(x_{i,s}|m_{i,s}) = \frac{1}{\sqrt{2\pi\alpha_s}} e^{\frac{-(x_{i,s}-m_{i,s})^2}{2\alpha_s}}$$

These likelihood functions are being sent by the coefficient nodes to the function nodes $g_{i,s}$. Each conditional function node $g_{i,s}$ is responsible to make a soft decision about whether its associated coefficient $x_{i,s}$ comes from the low-variance or the high-variance Gaussian distribution. This decision should be made from the provided likelihood functions of the upper part of the graph. They do this job by integrating the product of their own conditional density function $\Pr(x_{i,s}|P_{i,s})$ with the corresponding continuous likelihood function. So they essentially send the discrete likelihood probability vector of $\left[\begin{smallmatrix} p'_{i,s} \\ 1-p'_{i,s} \end{smallmatrix}\right]$ to $P_{i,s}$ according to the equation of $\mu_{g_{i,s}\to P_{i,s}}$ which describes the belief of the upper part of the $s-$th factor graph after the $t-$th iteration about whether or not the $i-$th index is in the common support set. Thus, each $P_{i,s}$ will receive a different likelihood probability vectors from each neighbor node.

If $h_{i,s}$ was the only source of prior information, then the messages from $g_{i,s}$ to the coefficient nodes would have remained the same throughout the algorithm. But here, in the case of jointly sparse signals, another source of information which comes from the spatial correlation between different sensor nodes is also available and this additional information is connected to the $P_{i,s}$ in the factor graph and contributes to the posterior probability of the $i-$th coefficient being in the support set. In this case, the messages from $g_{i,s}$ to the coefficient nodes at each sensor node will no longer remain the same. In fact, the product of all the incoming probability vectors to $P_{i,s}$ forms $\left[\begin{smallmatrix} p_{i,s} \\ 1-p_{i,s} \end{smallmatrix}\right]$ which is the belief of variable node $P_{i,s}$ from its own probability of being in the support set after the $t-$th iteration. Then the updated $p_{i,s}$ will be sent to $g_{i,s}$ and $g_{i,s}$ generates the updated compressible prior using $H(x_{i,s}) = p_{i,s}\mathcal{N}(0,\sigma_0^2) + (1-p_{i,s})\mathcal{N}(0,\sigma_1^2)$ and provides it to the upper part of the factor graph of each sensor. This new compressible prior defines new denoising functions F and G (discussed in the next section), and the messages that are being sent back and forth between the coefficient nodes and delta function nodes in the upper half of each factor graph will be computed according to the updated F and G functions. So essentially all the spatial correlation is incorporated in the compressible prior and we run one iteration of AMP for all sensors separately and then update the prior and this process will be repeated again until the algorithm converges. In this non-centralized network, we can prove the following Proposition.

Proposition 1: If the graph is connected and one of the sensors finds the true support set, then all of the other sensors in the network will achieve perfect reconstruction.

Proof: If node A finds the true support set at some iteration, the likelihood function that the coefficient nodes will receive from the upper part of the graph will be a delta function. In the next iterations, node A will receive different likelihood probabilities from its neighbors which together form a new prior for this node. But since the likelihood function of the node is a delta function, the support set of node A would not change regardless of the condition of other nodes and it always stays on the true support set. Thus, the likelihood probability that it transmits to a neighbor node such as B, is always a sequence of 0 and 1 which denotes the true support set. In the next iterations, node B multiplies all the incoming likelihood probabilities (including the one received from A) by its prior and forms a new prior. Thus, the new prior of node B will always be the true support set and thus regardless of its measurements, after some iterations, node B also achieves perfect reconstruction and this process happens again for the neighbors of node B. Since the graph is connected, after some iterations, the true support set will be propagated through the network and all the nodes achieve perfect reconstruction.

III. Approximating the Messages

In this work, we use the AMP framework [9] to approximate the messages. The idea is that in the large system limits, using the Central Limit Theorem, we can approximate the messages from delta function nodes to the coefficient nodes with Gaussian functions and the messages from the coefficient nodes to the delta function nodes could be approximated as the product of Gaussian and Gaussian mixture distributions. So instead of passing the real functions, we just pass the parameters of these distributions in our loopy belief propagation.

Let us assume that, the mean of the messages from the $i-$th coefficient to the $j-$th delta function at the $t-$th iteration for the $s-$sensor is $x^t_{i\to j,s}$ and its variance is $\beta^t_{i\to j,s}$. Then using the Central Limit Theorem, we can prove that $\mu^t_{f_{j,s}\to x_{i,s}}(x_{i,s}) \approx \mathcal{N}(\frac{z^t_{j\to i,s}}{\Phi_{ji,s}}, \frac{\alpha^t_{j\to i,s}}{\Phi^2_{ji,s}})$ where

$$z^t_{j\to i,s} = y_{j,s} - \sum_{\substack{k=1\\k\neq i}}^{N} \Phi_{jk,s} x^t_{k\to j,s} \text{ and } \alpha^t_{j\to i,s} = \sum_{\substack{k=1\\k\neq i}}^{N} \Phi^2_{jk,s}\beta^t_{k\to j,s}$$

The messages $\mu^{t+1}_{x_{i,s}\to f_{j,s}}(x_{i,s})$ can be approximated as the product of a Gaussian mixture density function with a Gaussian density function as

$$\mu^{t+1}_{x_{i,s}\to f_{j,s}}(x_{i,s}) \propto H(x_{i,s}) e^{\frac{-(x_{i,s}-m^t_{i\to j,s})^2}{2\alpha^t_s}}$$

where $m^t_{i\to j,s} = \sum_{\substack{k=1\\k\neq j}}^{M} \Phi_{ki,s} z^t_{k\to i,s}$ and it is assumed that $\alpha^t_{j\to i,s}$ is an edge-independent quantity ($\alpha^t_s = \alpha^t_{j\to i,s}$).

Let us assume the expectation of X_i, with respect to the distribution $\mu^{t+1}_{x_i\to f_j}(x_i)$, is the function $F(m,\alpha,p_{i,s})$ and the variance is $G(m,\alpha,p_{i,s})$. Then we have

$$x^{t+1}_{i\to j,s} = F(m^t_{i\to j,s}, \alpha^t_s, p^t_{i,s}) = F(\sum_{\substack{k=1\\k\neq j}}^{M} \Phi_{ki,s} z^t_{k\to i,s}, \alpha^t_s, p^t_{i,s})$$

$$\beta^{t+1}_{i\to j,s} = G(m^t_{i\to j,s}, \alpha^t_s, p^t_{i,s}) = G(\sum_{\substack{k=1\\k\neq j}}^{M} \Phi_{ki,s} z^t_{k\to i,s}, \alpha^t_s, p^t_{i,s})$$

$$\alpha^{t+1}_s = \frac{1}{M}\sum_{i=1}^{N} G(\sum_{j=1}^{M} \Phi_{ji,s} z^t_{j\to i,s}, \alpha^t_s, p^t_{i,s})$$

$$\mu^{t+1}_{x_{i,s}\to g_{i,s}}(x_{i,s}) \propto e^{\frac{-(x_{i,s}-m^t_{i,s})^2}{2\alpha^t_s}}$$

where $m^t_{i,s} = \sum_{j=1}^{M} \Phi_{ji,s} z^t_{j\to i,s}$. Furthermore, the messages from $g_{i,s}$ nodes to $P_{i,s}$ nodes can be approximated as

$$\mu^t_{g_{i,s}\to P_{i,s}}(P_{i,s}) = \left[\begin{smallmatrix} p'^t_{i,s} \\ 1-p'^t_{i,s} \end{smallmatrix}\right]$$

$$\Rightarrow p'^{t}_{i,s} = \frac{\frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_0^{2\,t})}} e^{\frac{-(m^t_{i,s})^2}{2(\alpha_s^t+\sigma_0^{2\,t})}}}{\frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_0^{2\,t})}} e^{\frac{-(m^t_{i,s})^2}{2(\alpha_s^t+\sigma_0^{2\,t})}} + \frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_1^{2\,t})}} e^{\frac{-(m^t_{i,s})^2}{2(\alpha_s^t+\sigma_1^{2\,t})}}}$$

But the above algorithm still needs passing of MN messages at each iteration and thus is computationally expensive. To address this problem, we utilize the first order approximation used in [9]. Therefore, the AMP algorithm for jointly sparse signals will be simplified as follows.

DCS-AMP Algorithm
1) Initialize: $\mathbf{x}_s^0 = 0$, $\mathbf{p}_s^0 = \mathbf{q}$, $\mathbf{z}_s^{-1} = 0$, $t = 0$
2) **while** $\|\mathbf{y}_s - \Phi_s \mathbf{x}_s^t\|_2 > \epsilon$ and $t < \#iterations$ **do**
3) $\mathbf{z}_s^t = \mathbf{y}_s - \Phi_s \mathbf{x}_s^t + \frac{N}{M}\mathbf{z}_s^{t-1}\langle F'(\mathbf{x}_s^{t-1} + \Phi_s^* \mathbf{z}_s^{t-1}, \alpha_s^{t-1}, \mathbf{p}_s^{t-1})\rangle$
4) $\mathbf{m}_s^t = \mathbf{x}_s^t + \Phi_s^* \mathbf{z}_s^t$
5) $\alpha_s^{t+1} = \frac{N}{M}\langle G(\mathbf{m}_s^t, \alpha_s^t, \mathbf{p}_s^t)\rangle$
6) $\mathbf{x}_s^{t+1} = F(\mathbf{m}_s^t, \alpha_s^t, \mathbf{p}_s^t)$
7)$\mathbf{p}'^{t}_s = \frac{\frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_0^{2\,t})}} e^{\frac{-(\mathbf{m}_s^t)^2}{2(\alpha_s^t+\sigma_0^{2\,t})}}}{\frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_0^{2\,t})}} e^{\frac{-(\mathbf{m}_s^t)^2}{2(\alpha_s^t+\sigma_0^{2\,t})}} + \frac{1}{\sqrt{2\pi(\alpha_s^t+\sigma_1^{2\,t})}} e^{\frac{-(\mathbf{m}_s^t)^2}{2(\alpha_s^t+\sigma_1^{2\,t})}}}$
8)$\begin{bmatrix} p^{t+1}_{i,s} \\ 1-p^{t+1}_{i,s} \end{bmatrix} \propto \begin{bmatrix} q_{i,s} \\ 1-q_{i,s} \end{bmatrix} \prod_{k\in N(s)} \begin{bmatrix} p'^{t}_{i,k} \\ 1-p'^{t}_{i,k} \end{bmatrix}$
9)$t \leftarrow t+1$
10) **end while**

Here $\langle \mathbf{x} \rangle$ denotes the average of the vector $\mathbf{x}$, F and G are applied element-wise on vectors, F' is the derivative of F with respect to the first argument and Φ^* is the transpose of Φ.

IV. Simulation Results

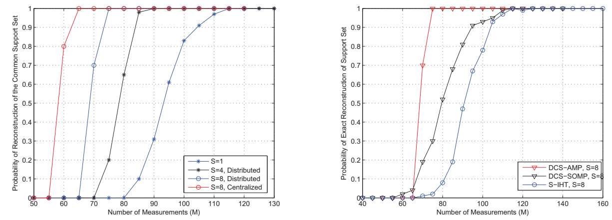
We consider signals of length $N = 200$ with the sparsity level of 50. We further assume $\sigma_0 = 0.1$ and $\sigma_1 = 1$. The noise is modeled by assuming a non-zero variance for σ_0.

A. Reconstruction Rate of DCS-AMP

In this part, we investigate the effect of the number of sensors on the reconstruction rate of the *common support set*. We run the DCS-AMP algorithm with 30 iterations on the signals with $S \in \{1, 4, 8\}$ in a network where node i is connected to the node $i + 1$ and the last node is connected to the first node. We also compare this case with the case of centralized network with $S = 8$. Fig. 2(a) indicates that by increasing the number of sensors, the algorithm does a better job. Also, from this Figure, we can see that the reconstruction rate of the ideal case (synchronous and fully connected network) is just 15% better than that of non-centralized mode which shows that in the long run, the non-centralized mode can incorporate most the diversity of the spatial information of the network in reconstruction.

B. Comparison of DCS-AMP with SOMP and S-IHT

We now present a simulation comparing the DCS-AMP algorithm with two other algorithms in distributed compressed sensing, so called DCS-SOMP [5] and S-IHT [11] for the case of $S = 8$ signals. Fig. 2(b) plots the probability of exact reconstruction of the *support set* versus the number of measurements per signal (M). We can see that the performance of our algorithm even in the non-centralized mode outperforms the reconstruction rate of both of the algorithms.



(a) Reconstruction of the common support in DCS-AMP (b) Comparison of DCS-AMP with SOMP and S-IHT

Fig. 2. Performance of DCS-AMP, SOMP and S-IHT

V. Conclusion

In this work, we derived a fast iterative algorithm to reconstruct a set of jointly sparse signals in a cognitive radio network. Our method could be adopted in ad-hoc networks where there is no fusion center. We showed that the proposed DCS-AMP algorithm could capture spatial diversity better than the other joint sparse recovery algorithms such as DCS-SOMP or S-IHT.

References

[1] D. Cabric, A. Tkachenko, and R. W. Brodersen, “Spectrum sensing measurements of pilot, energy, and collaborative detection,” in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1–7, IEEE, 2006.

[2] V. Havary-Nassab, S. Hassan, and S. Valaee, “Compressive detection for wide-band spectrum sensing,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010.

[3] Y. Wang, A. Pandharipande, Y. L. Polo, and G. Leus, “Distributed compressive wide-band spectrum sensing,” in *Information Theory and Applications Workshop, 2009*, pp. 178–183, IEEE, 2009.

[4] J. Tropp, A. Gilbert, and M. Strauss, “Simultaneous sparse approximation via greedy pursuit,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on*, vol. 5, pp. v/721 – v/724 Vol. 5, 2005.

[5] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, “Distributed compressed sensing of jointly sparse signals,” in *In Asilomar Conf. Signals, Sys., Comput*, pp. 1537–1541, 2005.

[6] K. Lee, Y. Bresler, and M. Junge, “Subspace methods for joint sparse recovery,” *Information Theory, IEEE Transactions on*, vol. 58, no. 6, pp. 3613–3641, 2012.

[7] M. Mishali and Y. Eldar, “Reduce and boost: Recovering arbitrary sets of jointly sparse vectors,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4692–4702, 2008.

[8] J. Ziniel and P. Schniter, “Efficient high-dimensional inference in the multiple measurement vector problem,” 2011.

[9] A. Maleki and A. Montanari, “Analysis of approximate message passing algorithm,” in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pp. 1–7, IEEE, 2010.

[10] D. Donoho, A. Maleki, and A. Montanari, “How to design message passing algorithms for compressed sensing,” 2010.

[11] A. Makhzani and S. Valaee, “Reconstruction of jointly sparse signals using iterative hard thresholding,” in *Communications (ICC), 2012 IEEE International Conference on*, IEEE, 2012.

[12] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, “Beyond nyquist: Efficient sampling of sparse bandlimited signals,” *Information Theory, IEEE Transactions on*, vol. 56, no. 1, pp. 520–544, 2010.