A Robust Algorithm for Joint-Sparse Recovery

Md Mashud Hyder and Kaushik Mahata

***Abstract*—We address the problem of finding a set of sparse signals that have nonzero coefficients in the same locations from a set of their compressed measurements. A mixed $\ell_{2,0}$ norm optimization approach is considered. A cost function appropriate to the joint-sparse problem is developed, and an algorithm is derived. Compared to other convex relaxation based techniques, the results obtained by the proposed method show a clear improvement in both noiseless and noisy environments.**

***Index Terms*—Basis pursuit, compressive sampling, joint-sparse, multiple measurement vectors, sparse representation.**

I. Introduction

A signal $x \in \mathbb{R}^n$ is called k-sparse if it has at most $k < n$ nonzero components. The recent research in compressed sensing has revealed that a k-sparse x can be uniquely recovered from a 'measurement' $y \in \mathbb{R}^m$, $m < n$, of the form

$$y = \Phi x$$

where $\Phi \in \mathbb{R}^{m \times n}$ is a known measurement matrix [1]. Two conditions are needed to ensure that x can be recovered from y uniquely by using ℓ_0 minimization: i) $k \leq m/2$ and ii) every m columns of Φ form a basis of $\mathbb{R}^m$. Like the recovery of sparse signals, another closely related problem is getting increased attention of the research community. Here, one is interested in recovering a *jointly row sparse* matrix $X_0 \in \mathbb{R}^{n \times r}$ from Y formed by multiplying X_0 by Φ

$$Y = \Phi X_0. \tag{1}$$

By the term "jointly row sparse" we mean that only k rows of X_0 are nonzero. The problem of recovering jointly sparse matrices will be called the multiple measurement vector (MMV) problem, which arises naturally in source localization, neuromagnetic imaging [2], and equalization of sparse communication channels [3], and several other applications [4]. The following lemma [2] gives the sufficient conditions for the existence of the unique solution to the MMV problem.

Lemma 1: Let rank $(Y) = r \leq m$, and every m columns of Φ forms a basis of $\mathbb{R}^m$. Then a solution to (1) with k nonzero rows is unique provided that $k \leq \lceil (m+r)/2 \rceil - 1$($\lceil . \rceil$ denotes the ceiling operation).

Manuscript received June 03, 2009; revised July 02, 2009. First published July 21, 2009; current version published September 23, 2009. This work was supported by the Australian Research Council. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tan F. Wong.

The authors are with the Department of Electrical Engineering, University of Newcastle, New South Wales 2308, Australia (e-mail: md.hyder@studentmail.newcastle.edu.au; Kaushik.Mahata@newcastle.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2009.2028107

Researchers spanning a diverse range of viewpoints have advocated mixed-norm minimization approach to solve the MMV problem. Here, one solves

$$\min_{X \in \mathbb{X}} \|X\|_{p,q}, \quad \mathbb{X} := \{X \in \mathbb{R}^{n \times r} : Y = \Phi X\} \tag{2}$$

for various combinations of p and q, where the mixed norm $\|X\|_{p,q}$ is defined as

$$\|X\|_{p,q} = \left(\sum_{j=1}^{n} \left(\sum_{i=1}^{r} X[j,i]^p \right)^{q/p} \right)^{1/q}.$$

Cotter *et al.* [2] use $p = 2, q \leq 1$, Tropp [5] analyzes for $p = 1, q = \infty$ and Elder *et al.* [6] use $p = 2, q = 1$. In [6] the sufficient conditions for MMV recovery by using $p = 2$ and $q = 1$ are presented. Berg *et al.* [4] analyze some properties when $p = q = 1$ and $p = 2, q = 1$. In a broad sense, all these algorithms offer some form of convex relaxation to the case $p = 2, q = 0$, which guarantees unique recovery under the assumptions of Lemma 1. Unfortunately, this case of $p = 2, q = 0$ leads to a nonconvex problem which is not tractable. ReMBo [7], on the other hand, reduces MMV to a series of SMV problems. ReMBo proceeds by taking a random vector $w \in \mathbb{R}^r$ and combining the individual observations in Y into a single weighted observation $y := Yw$. Subsequently, it solves a single measurement vector problem $\Phi x = y$ by using a suitable algorithm and checks if the computed solution $\overline{x}$ is sufficiently sparse. If not, the above steps are repeated with a different w; the algorithm stops when a maximum number of trials is reached. Once the ReMBo algorithm has found a sufficiently sparse solution it solves a least-squares problem using only those columns in Φ corresponding to the support.

Most the proposed methods achieve a good recovery rate for small values of k. However, to the best of our knowledge, it has not been possible to achieve good recovery rate when k is close to the upper bound provided by Lemma 1. This in some applications can lead to limitations. For example, in source separation application, k denotes the number of sources that a fixed sensor array can handle, and being able to recover signals with larger k leads to a significant improvement. Another important issue is the performance of the algorithms in presence of measurement noise. In the numerical simulations it was found that most of the algorithms suffer from performance degradation in presence of noise. In order to cope with these issues, we advocate using $p = 2$ and $q = 0$ in (2). We extend the zero-norm approximation algorithm [8]. The new algorithm is called JLZA. Simulation results indicate that the proposed approach does achieve the theoretical limit on k given by Lemma 1. In addition, it is fast, and robust to the measurement noise. Furthermore, when r increases, the complexity of JLZA remains almost constant.

II. Joint $\ell_{2,0}$ Approximation Algorithm (JLZA)

In the following, we develop an algorithm to solve (2) with an approximation to $\|X\|_{2,0}$. One can write $\|X\|_{2,0}$ as

$$\|X\|_{2,0} = \sum_{j=1}^{m} I(\|X[j,:]\|_2)$$

where the ith row of a matrix A is denoted by $A[i,:]$, and I is the indicator function

$$I(\alpha) = \begin{cases} 1, & \alpha > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Since $\|X\|_{2,0}$ is nonsmooth, we work with an approximation of $\|X\|_{2,0}$ using Gaussian functions [9]. Let

$$f_\sigma(\alpha) = \mathrm{e}^{-\alpha^2/2\sigma^2}. \tag{4}$$

Clearly $\lim_{\sigma\to 0} f_\sigma(\alpha) = 1 - I(\alpha)$. Consequently, the function

$$F_\sigma(X) = \sum_{t=1}^{n} f_\sigma(\|X[t,:]\|_2)$$

behaves like $n - \|X\|_{2,0}$ when $\sigma \to 0$. This motivates the following approximate way of reformulating (2) for $p = 2$ and $q = 0$:

$$X_* = \arg\max_{X\in\mathbb{X}} F_\sigma(X). \tag{5}$$

However, $F_\sigma(X)$ has many local maxima for small values of σ. Nevertheless, as σ increases, $F_\sigma(X)$ becomes smoother, and for a sufficiently large σ, one has $X_* = \Phi'(\Phi\Phi')^{-1}Y$ [9]. Hence, the standard procedure is to take a large σ initially and solve (5). Subsequently, σ is reduced by some small factor and (5) is solved again. The procedure is repeated until a convergence criterion is satisfied.

Algorithm Derivation

The Lagrangian $L(X,\Upsilon)$ for (5) is

$$L(X,\Upsilon) = F_\sigma(X) + \mathrm{Trace}\{\Upsilon'(\Phi X - Y)\}. \tag{6}$$

Let us define the matrix $\partial L/\partial X$, such that $[\partial L/\partial X]_{ij} = \partial L/\partial X[i,j]$, where $X[i,j]$ denotes the element of X at its ith row and j-th column. Now (5) implies that the stationary point (X_*, Υ_*) of $L(X,\Upsilon)$ satisfies,

$$\begin{aligned} \frac{\partial L(X_*,\Upsilon_*)}{\partial X} &= \frac{\partial F_\sigma(X_*)}{\partial X} + \Phi'\Upsilon* = 0, \\ \frac{\partial L(X_*,\Upsilon*)}{\partial \Upsilon} &= \Phi X_* - Y = 0. \end{aligned} \tag{7}$$

Also, it is readily verified that

$$\begin{aligned} \left[\frac{\partial F_\sigma(X)}{\partial X}\right]_{ij} &= -\frac{f_\sigma(\|X[i,:]\|_2)}{\sigma^2} X[i,j] \\ \Rightarrow \frac{\partial F_\sigma(X)}{\partial X} &= -\frac{1}{\sigma^2} W(X)X \end{aligned} \tag{8}$$

where $W(X) = \mathrm{diag}\{f_\sigma(\|X[1,:]\|_2), \ldots, f_\sigma(\|X[n,:]\|_2)\}$. Then (7)–(8) gives

$$X_* = \sigma^2 W^{-1}(X_*)\Phi'\Upsilon_*. \tag{9}$$

TABLE I
JLZA Algorithm

Initialization
1. Set $X^{(0)} = \Phi'(\Phi\Phi')^{-1}Y$.
2. Set $\sigma = 1$ and $\rho, \eta, \gamma \in \{0, 1\}$.

repeat
3. Set $\lambda = 1$.
4. while $F_\sigma(\lambda\zeta(X^{(i)}) + (1-\lambda)X^{(i)}) < F_\sigma(X^{(i)})$
   $\lambda = \gamma\lambda$.
   end
5. $X^{(i+1)} = \lambda\zeta(X^{(i)}) + (1-\lambda)X^{(i)}$
6. If $\tau^{(i)} = \|X^{(i+1)} - X^{(i)}\|_2 < \eta\sigma$ then $\sigma = \rho\sigma$.

until the stopping criterion is satisfied.

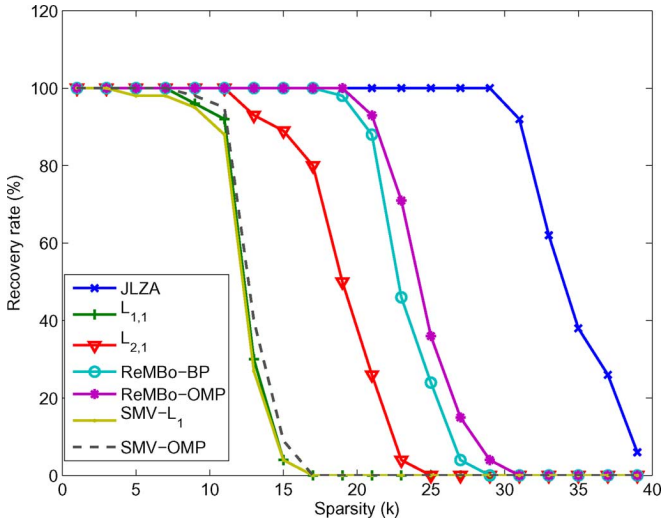


Fig. 1. Recovery rates (averaged over 100 trials) for randomly generated $X_0 \in \mathbb{R}^{128\times 10}$ with different sparsity k. The value of m is fixed to 50 and $r = 10$.

Substitute this for X_* in the second equation of (7), then solve for Υ_*, and substitute the solution to Υ_* in (9) to get

$$X_* = W^{-1}(X_*)\Phi'\left[\Phi W^{-1}(X_*)\Phi'\right]^{-1} Y. \tag{10}$$

Equation (10) is nonlinear, and it can be used in a fixed-point iteration. The following Lemma indicates that if a certain condition holds then the algorithm will converge to a local maxima.

Lemma 2: Let us define the map $\zeta : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ such that

$$\zeta(X) = W^{-1}(X)\Phi'\left[\Phi\, W^{-1}(X)\Phi'\right]^{-1} Y. \tag{11}$$

Then $\Phi\zeta(X) = Y$. Let $X \in \mathbb{R}^{n\times r}$ such that $\Phi X = Y, \zeta(X) \neq X$, and

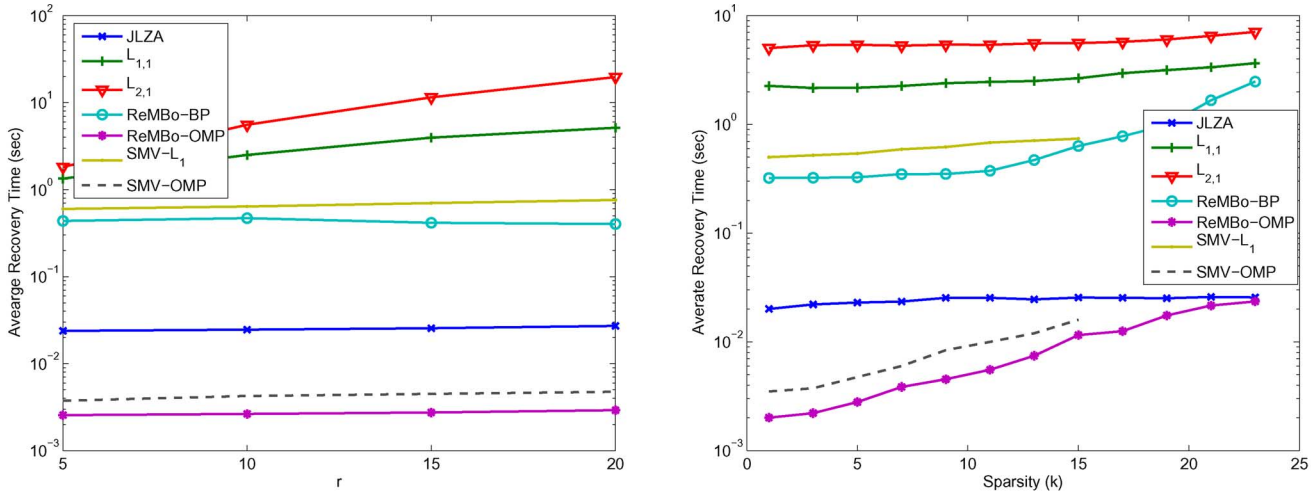
$$\left[\frac{\partial F_\sigma(X)}{\partial X}\right] \neq 0. \tag{12}$$

Then there exists λ satisfying $0 < \lambda \leq 1$ such that

$$F_\sigma\left\{\lambda\zeta(X) + (1-\lambda)X\right\} > F_\sigma(X).$$

□

Lemma 2 is a generalization of Lemma 1 in [8].



(a) Average recovery time (on 100 runs) versus r for fixed $k = 10$ (b) Average recovery time (on 100 runs) versus k for fixed $r = 10$

Fig. 2. Average recovery time of various MMV techniques. The value of $n = 128$ and $m = 50$.

Algorithm

The final algorithm is given in Table I. In the algorithm, $X^{(i)}$ denotes the value of X updated at i^{th} iteration. We start with X_* for $\sigma \to \infty$, [9] and set $\sigma = 1$ for the first iteration. For maximizing F_σ, we use $\zeta(X) - X$ as ascent-direction. By choosing a proper factor λ i.e., $0 < \lambda < 1$, the algorithm can be accelerated to converge the optimal solution rapidly. However, in our extensive experimental study we have noticed that $\lambda = 1$ is sufficient, backtracking (step 3–4) is not needed. We fixed γ to 0.5. The inner-iteration for maximizing F_σ for a given σ terminates when $\tau^{(i)} < \eta\sigma$, see step 6. A wide range of numerical experiments suggests that the best setting of η is 0.5. The stopping criterion of JLZA is based on a small σ denoted by σ_0. Upon convergence of each inner iteration we lower σ by a factor ρ (step 6). When σ reaches to σ_0, JLZA stops its iterations. The final value of σ_0 depends on the noise level. For noiseless sparse recovery, ρ is chosen to 0.1 and σ_0 is fixed near to zero ($\sigma_0 = 10^{-4}$). However, in the noisy case, σ_0 should be left at some larger value as the system can not approximate the optimal X exactly and the solution fluctuates randomly. A wide range of numerical simulations in noisy cases suggests that the best setting of $\rho = 0.3$ and $\sigma_0 = 0.05$. Note that calculating ζ in step 4 requires inverting $W(X^{(i)})$ [see (10)]. However, $W(X^{(i)})$ may be close to singular in some cases. To make this situation tractable, we use $W(X^{(i)}) + 10^{-8} I_n$ instead of $W(X^{(i)})$.

III. Simulation Results

Two types of experiments are presented: exact signal recovery in noiseless case (Figs. 1–2) and approximate signal recovery in noisy environment (Figs. 3–4). We choose $n = 128$ and $m = 50$. For experimental setup, we consider the procedure of [7]. The following steps are repeated 100 times for each experiment. i) We construct $\Phi \in \mathbb{R}^{50\times 128}$ consisting of iid Gaussian random variables with zero mean and variance one, ii) for every sparsity (k), we generate a k row sparse matrix $X_0 \in \mathbb{R}^{n\times r}$. The nonzero location set is drawn uniformly at random among $\binom{n}{k}$ choices. The nonzero elements are iid Gaussian random variables, as in (i). We compare the performance of JLZA to $\ell_{1,1}$, $\ell_{2,1}$ and ReMBo. We also decompose the MMV problem into a series of SMV problems. That is we consider every column of Y independently, which generates r SMV problems. We then solve them independently by using ℓ_1 (SMV-ℓ_1) and orthogonal matching pursuit (SMV-OMP) algorithms. We use SDPT3 [10] through the CVX interface [11] for ReMBo, $\ell_{1,1}$ and $\ell_{2,1}$[1]. In the ReMBo algorithm, the components of $w \in \mathbb{R}^r$ (see Section I) are iid random variables with uniform distribution on $[-1, 1]$, while the associated SMV problem is solved by using both basis pursuit (ReMBo-BP) and orthogonal matching pursuit (ReMBo-OMP). In orthogonal matching pursuit, we stop computation when the second norm of residual error becomes smaller than 10^{-16}. We allow a maximum of ten iterations to recover X_0 using different w.

In the noiseless cases, we say X_0 is to be recovered when the reconstructed X_* satisfies $||X_0 - X_*||_\infty < 10^{-5}$. In the noisy environment, the objective function of the convex relaxation based approaches [(2)] is modified as

$$\min_X ||X||_{p,q} \text{ subject to } ||Y - \Phi X||_F \le \mu \qquad (13)$$

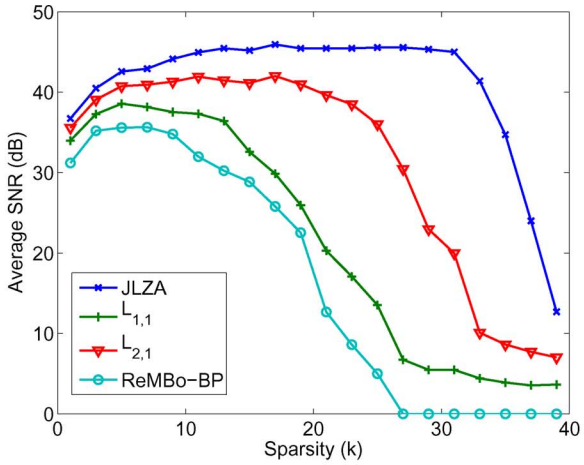
where μ^2 is the variance of the noise signal. The similar model is used for the SMV solver used for ReMBo. The SNR is used to measure performance. SNR is defined as

$$SNR(X_*) = \frac{1}{r}\sum_{i=1}^{r} 20\log\left(\frac{||X_*[:,i]||_2}{||X_*[:,i] - X_0[:,i]||_2}\right).$$

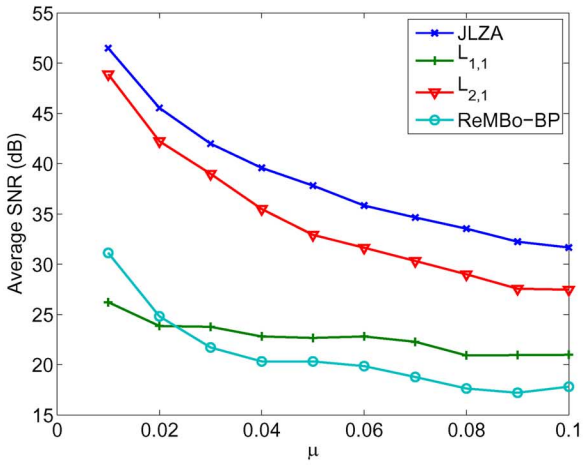
Our simulations are performed in MATLAB7 environment using an Intel Core 2 Duo, 2.66-GHz processor with 2 GB of memory, under Mac OS X Version 10.5.5 operating system.

Fig. 1 shows the recovery rate as a function of k for various techniques in noise-less case. Note that JLZA achieves a very high recovery rate for $k \le 29$, and 29 happens to be the upper bound given by Lemma 1. Note in Fig. 2 that $\ell_{1,1}$ and $\ell_{2,1}$ require large recovery time compared to ReMBo. Note that while the recovery time of ReMBo is not affected by r, it increases quickly with k (i.e., $k \ge 15$). The average recovery time of

[1]Source code at http://www.cs.ubc.ca/~mpf/?n=JointSparse



(a) Average SNR (on 100 runs) versus k for fixed $r = 10$ and $\mu = 0.03$



(b) Average SNR (on 100 runs) versus μ for fixed $k = 20$ and $r = 10$.

Fig. 3. Average SNR achieved by various MMV techniques. The value of $n = 128$ and $m = 50$.

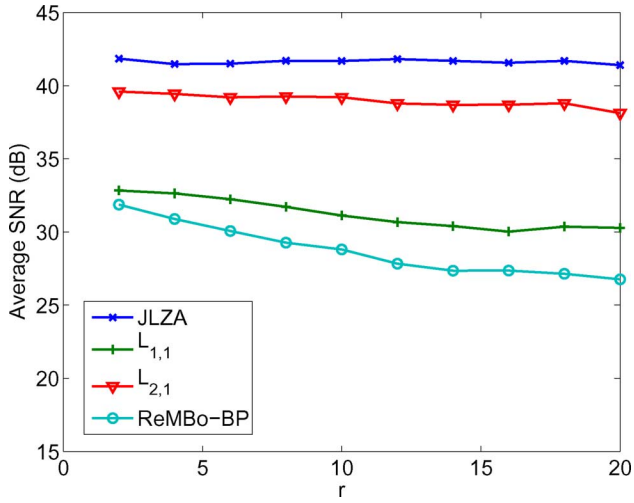


Fig. 4. Average SNR (on 100 runs) versus r for fixed $k = 15$, $\mu = 0.03$, $n = 128$ and $m = 50$.

JLZA is not affected by r and k, which, however, will increase with increasing m. This is because the evaluation of $\zeta(X)$ in (11) depends on m.

The performance of different algorithms in noisy environment is investigated in Figs. 3–4. A model including additive noise can be written as $Y = \Phi X + E$, where E consists of iid Gaussian random variables with mean zero and variance μ^2. RemBo-OMP, SMV-ℓ_1 and SMV-OMP perform poorly with noisy data, and results are not shown. As opposed to convex optimization methods, JLZA demonstrates an improved recovery rate in noisy environment. Fig. 3(a) shows the level of sparsity needed to estimate a noisy signal from a fixed measurement. Observe that JLZA enjoys a clear superiority over other algorithms. Also, for each algorithm there is a critical level of sparsity. When sparsity increases beyond this level, the performance drops rapidly. The level is 15, 18, 25, and 32 for ReMBo-BP, $\ell_{1,1}$, $\ell_{2,1}$ and JLZA, respectively. Fig. 3(b) demonstrates the effect of μ on joint-sparse signal for different algorithms. Finally, Fig. 4 provides a comparison of SNR archived by different algorithms as a function of r, showing the improvement achieved by JLZA.

IV. Conclusion

An $\ell_{2,0}$ approximation algorithm is applied in joint sparse multiple measurement vectors problem. In contrast to other convex relaxation algorithms, the proposed algorithm shows a clear performance improvement. Furthermore, the empirical results suggest that the proposed algorithm can achieve the theoretical limit of sparsity k with very high probability.

References

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

[2] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[3] S. Cotter and B. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, pp. 374–377, Mar. 2002.

[4] E. van den Berg and M. Friedlander, Joint-Sparse Recovery From Multiple Measurements 2009, arXiv:0904.2051.

[5] J. Tropp, "Algorithms for simultaneous sparse approximation: Part ii: Convex relaxation," *Signal Process.*, vol. 86, no. 3, pp. 589–602, 2006.

[6] Y. C. Eldar and M. Mishali, Robust Recovery of Signals From a Structured Union of Subspaces 2008 [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:0807.4581

[7] M. Mishali and Y. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4692–4702, Oct. 2008.

[8] M. Hyder and K. Mahata, "An approximate l0 norm minimization algorithm for compressed sensing," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 3365–3368.

[9] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[10] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathemat. Programm.*, vol. 95, no. 2, pp. 189–217, Feb. 2003.

[11] M. Grant and S. Boyd, Cvx: Matlab Software for Diciplined Convex Programming 2009 [Online]. Available: http://stanford.edu/~boyd/cvx