A Fast Algorithm for Recovery of Jointly Sparse Vectors based on the Alternating Direction Methods

Hongtao Lu **Xianzhong Long** **Jingyuan Lv**
MOE-MS Key Laboratory for Intelligent Computing and Intelligent Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200240, China
{*htlu,lxz85,lwjingyuan*}*@sjtu.edu.cn*

Abstract

The standard compressive sensing (CS) aims to recover sparse signal from single measurement vector which is known as SMV model. By contrast, recovery of sparse signals from multiple measurement vectors is called MMV model. In this paper, we consider the recovery of jointly sparse signals in the MMV model where multiple signal measurements are represented as a matrix and the sparsity of signal occurs in common locations. The sparse MMV model can be formulated as a matrix $(2,1)$-norm minimization problem, which is much more difficult to solve than the l_1-norm minimization in standard CS. In this paper, we propose a very fast algorithm, called MMV-ADM, to solve the jointly sparse signal recovery problem in MMV settings based on the alternating direction method (ADM). The MMV-ADM alternately updates the recovered signal matrix, the Lagrangian multiplier and the residue, and all update rules only involve matrix or vector multiplications and summations, so it is simple, easy to implement and much faster than the state-of-the-art method MMVprox. Numerical simulations show that MMV-ADM is at least dozens of times faster than MMVprox with comparable recovery accuracy.

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

1 INTRODUCTION

Many signals of interest often have sparse representations, meaning that signal is well approximated by only a few nonzero coefficients in a specific basis. Compressive sensing (CS) has recently emerged as an active research area which aims to recover sparse signals from measurement data (Candes et al., 2006; Donoho, 2006a). In the basic CS, the unknown sparse signal is recovered from a single measurement vector, this is referred to as a single measurement vector (SMV) model. In this paper, we consider the problem of finding sparse representation of signals from multiple measurement vectors, which is known as the MMV model (Chen et al., 2006). In the MMV model, signals are represented as matrices and are assumed to have the same sparsity structure. Specifically, the entire rows of signal matrix may be 0.

The MMV model was initially motivated by a neuromagnetic inverse problem that arises in Magnetoencephalography (MEG), a brain imaging modality (Cotter et al., 2005). It is assumed that MEG signal is a mixture of activities at a small number of possible activation regions in the brain. MMV model has also been found in array processing (Gorodnitsky et al., 1997), nonparametric spectrum analysis of time series (Stoica et al.,1997), equalization of sparse communication channel, linear inverse problem (Cotter et al.,2005), DNA microarrays (Erickson et al., 2008) and source location in sensor networks (Malioutov et al.,2003) etc..

Some known theoretical results of SMV have been generalized to MMV (Chen at al., 2006) such as the uniqueness under both l_0-norm criteria and l_1-norm criteria, and the equivalence between the l_0-norm approach and the l_1-norm criteria. Several computation algorithms have also been proposed for solving MMV problem. It has been proven that under certain conditions, the orthogonal matching pursuit (OMP) method can find the sparsest solution for MMV, just

like in SMV (Chen et al.,2006). The matching pursuit (MP) algorithm and the FOCal Underdetermined System Solver (FOCUSS) for SMV have also been extended to MMV (Cotter et al.,2005; Gorodnitsky et al.,1997). Tropp et al. proposed the simultaneous orthogonal matching pursuit (S-OMP) method (Tropp et al., 2006a) and the convex relaxation method (Tropp, 2006b) to solve MMV.

Among these algorithms, most approaches adopted the matrix $(2,1)$-norm as the optimization objective functions (Chen et al.,2006; Cotter et al.,2005; Eldar et al.,2009). However, unlike the SMV model where the l_1 minimization can be efficiently solved, the convex optimization problem resulted from MMV is much more difficult to solve. Some existing algorithms reformulated it as a second-order cone programming (SOCP) problem or a semidefinite programming problem (SDP) (Stojnic et al.,2009), however both SOCP and SDP are computationally expensive and can only be used for small-size problems.

Recently, Sun et al. proposed a new algorithm called MMVprox to solve MMV problems (Sun et al.,2009). They first derived the dual optimization problem of the primal $(2,1)$-norm minimization problem, then formulated the dual optimization as a minmax problem, and finally employed the prox-method (Nemirovski, 2005) for solving variational inequality problems to solve the minmax problem. By taking advantage of the almost dimension-independent convergence rate $O(1/t)$ of the prox-method, the authors claimed that the MMVprox algorithm scales well to larger size problems. However, numerical simulations show that although MMVprox is much faster than SOCP, it is still very slow for relative large size problems, and its scalability is still not applicable for real large size problems in practice.

In this paper, we present a new fast algorithm (called MMV-ADM) to solve MMV problems based on the alternating direction method (ADM). ADM is an approximate augmented Lagrangian multiplier method and has recently been successfully applied to solve l_1-norm minimization problems in SMV model of compressive sensing (J.Yang et al.,2010). J. Yang et al. proposed several ADM based algorithms (called YALL1) and compare them with the state-of-the-art l_1-minimization algorithms. A.Yang et al. (2010) also made similar comparisons, and they pointed out that YALL1 is the fastest algorithm and has the best overall performance for synthetic data among the other state-of-the-art methods such as the orthogonal matching pursuit (OMP) (Donoho et al., 2006), homotopy method (Osborne et al., 2000), sparse reconstruction by separable approximations (SpaRSA) (Wright et al.,2009), fast iterative shrinkage-threshold algorithm (FISTA) (Beck et al., 2009) and L1LS (Kim et al.,2007).

Our works in this paper is to extend the ADM to solve the matrix $(2,1)$-norm minimization in the MMV model. Since the matrix $(2,1)$-norm is not componentwise separable as the vector l_1-norm, as pointed out in Sun et al. (2009), solving $(2,1)$-norm minimization is much more difficult than solving l_1-norm minimization. The proposed MMV-ADM only involves matrix or vector multiplication and summation operations, no need to evaluate first derivatives of the objective function at each iteration as in MMVprox algorithm, it is much faster than MMVprox. Numerical simulations show than MMV-ADM is dozens of times faster than MMVprox, so it scales well to large size problems.

Notations. For a d-dimensional vector $v = (v_i)_{i=1}^d$, its l_p-norm ($p \geq 1$) is defined as $\|v\|_p = (\sum_{i=1}^d v_i^p)^{1/p}$, for an $m \times d$ matrix $A = (a_{ij}) \in R^{m\times d}$, the (r,s)-norm of A is defined as $\|A\|_{r,s} = (\sum_{i=1}^m \|a^i\|_r^s)^{1/s}$, where a^i denotes the ith row of A. In particular, the $(2,1)$-norm is $\|A\|_{2,1} = \sum_{i=1}^m \|a^i\|_2$. The Frobenius norm $\|A\|_F$ is defined as $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$, it is easy to see that $\|A\|_F^2 = \langle A, A\rangle = \mathrm{tr}(A^T A)$, where $\langle\cdot,\cdot\rangle$ denotes the inner product of two matrices, $\mathrm{tr}(\cdot)$ is the trace of a matrix, A^T is the transpose of A.

2 PROBLEM FORMULATION

In the SMV model of CS, it is assumed that there exists a d-dimensional unknown sparse signal $x \in R^d$ and a measurement vector $b \in R^m$, they are related by a linear sensing process $b = Ax$, where $A \in R^{m\times d}$ is the sensing matrix and is assumed to be known. If we assume the sensing matrix A to be overcomplete, i.e., $m < d$, the original signal x can then be reconstructed from the underdetermined system of linear equations $Ax = b$, and there are infinite number of solutions for x. In order to recover the sparse signal x from this underdetermined linear system, in the basic CS, the approach is to seek the sparsest solution within all solutions of the linear system, i.e., via solving the following optimization problem

$$(\mathrm{p0}): \quad \min_x \|x\|_0, \quad \text{s.t.} \quad Ax = b \tag{1}$$

where $\|x\|_0$ denotes the l_0 (quasi) norm of x, i.e., the number of nonzero components of x. Since $\|x\|_0$ is not convex and is combinatorial in nature, l_0 minimization problem (p0) is NP-hard. It is thus often relaxed to the following convex l_1-norm minimization problem

$$(\mathrm{p1}): \quad \min_x \|x\|_1, \quad \text{s.t.} \quad Ax = b \tag{2}$$

Theoretical results show that under some mild conditions, the solution of (p1) is also the sparsest solution, i.e., the solution of (p0) (Donoho,2006b).

The l_1-minimization problem (p1) has been well studied, many efficient algorithms have been proposed to solve it, such as the homotopy method, OMP method, fast iterative-shrinkage thresholding and the ADM method mentioned above.

Unlike the SMV model, in the MMV model, n multiple measurement vectors are available which can be arranged as a measurement matrix $B \in R^{m\times n}$. The recovered signals are represented by a matrix $X \in R^{d\times n}$, the aim of MMV compressive sensing is to recover jointly sparse X from the system of the linear equations $AX = B$. In this paper, joint sparsity specifically means than X has many zero rows. This problem can be formulated as follows

$$(\text{P0}): \quad \min_X \|X\|_{2,0}, \quad \text{s.t.} \quad AX = B \tag{3}$$

where $\|X\|_{2,0}$ measures the number of rows in X that contain nonzero entries. However, (P0) is a combinatorial optimization problem and is thus NP-hard (Chen et al.,2006). Similar to the use of l_1-norm to replace l_0-norm in SMV, the matrix $(2,0)$-norm in (P0) is usually replaced by the matrix $(2,1)$-norm that results in the following convex relaxation problem

$$(\text{P1}): \quad \min_X \|X\|_{2,1}, \quad \text{s.t.} \quad AX = B \tag{4}$$

Existing algorithms reformulate (P1) as SOCP or SDP problem, and then solve it using the interior point method or the bundle method. However, being the second-order methods, the SOCP and SDP have high computation complexity, making them do not scale well to high-dimensional problems.

Recently, Sun et al. considered an equivalent optimization problem of (P1) by replacing the $(2,1)$-norm with its square, and obtain the following optimization

$$\min_X \frac{1}{2}\|X\|_{2,1}^2, \quad \text{s.t.} \quad AX = B \tag{5}$$

They then proposed an algorithm called MMVprox to solve (5). They first derived the dual problem of (5) as follows (Sun et al.,2009)

$$\max_Y \{ -\frac{1}{2}\|A^T Y\|_{2,\infty}^2 + < Y, B > \} \tag{6}$$

and then transformed (6) into a minmax optimization problem and finally solve it by the prox-method developed in (Nemirovski, 2005). The prox-method has been proven to enjoy almost dimension-independent convergence rate of $O(1/t)$, so the MMVprox algorithm is expected to be much faster than SOCP and SDP in the computation of sparse signal recovery of the MMV model. Experimental results in (Sun et al.,2009) confirmed this conclusion.

However, numerical experiments show that MMVprox is still very slow for relative large size problems. In this paper, we develop a new fast algorithm called MMV-ADM to efficiently solve the optimization problem (4) based on the alternating direction method (ADM). Our motivation is that among the state-of-the-art algorithms for solving the l_1-minimization problem (2), the well-known iterative shrinkage-thresholding algorithm (ISTA) has a worst-case convergence rate of $O(1/t)$ that is the same as the prox-method, the fast iterative shrinkage-thresholding algorithms (FISTA) developed in (Beck et al.,2009) has improved convergence rate of $O(1/t^2)$, better than the prox-method. However, A. Yang et al. have pointed out that ADM based method is even faster than FIST (A.Yang et al., 2010). So our MMV-ADM is expected to be much faster than MMVprox, numerical simulations in the experimental section confirm this.

3 RECOVERY OF JOINTLY SPARSE VECTORS BASED ON ADM

The classical alternating direction method (ADM) was designed to solve the following structured optimization problem (J.Yang et al.,2010)

$$\min_{x,y}\{f(x) + g(y)\} \quad \text{s.t.} \quad Ax + By = b \tag{7}$$

where x, y are vectors of dimension m and n, respectively, and f, g are two real-valued functions defined on m and n dimensional spaces, respectively. A, B, b are matrices or vectors of appropriate dimensions. Variables x and y are separate in the objective function and coupled only in the constraint. The augmented Lagrangian function of (7) is

$$L(x, y, \lambda) = f(x) + g(y) - \lambda^T(Ax + By - b) + \frac{\beta}{2}\|Ax + By - b\|_2^2 \tag{8}$$

where λ is the Lagrangian multiplier and $\beta > 0$ is a penalty parameter. Given $\lambda^{(k)}$, the value of *lambda* at the kth iteration, the classical augmented Lagrangian method iterates as follows

$$\begin{cases} (x^{(k+1)}, y^{(k+1)}) = \text{argmin}_{x,y} L(x, y, \lambda^{(k)}) \\ \lambda^{(k+1)} = \lambda^{(k)} - \gamma\beta(Ax^{(k+1)} + By^{(k+1)} - b) \end{cases} \tag{9}$$

When $\gamma \in (0, 2)$, convergence of the above iteration is guaranteed provided that the subproblem is solved with an increasingly high accuracy at every iteration (Rockafellar, 1973). However, accurate joint minimization with respect to (x, y) is usually computationally expensive. Instead, ADM employs the separability structure in (7) and replaces the joint minimization by

two simpler subproblems. Given $(y^{(k)}, \lambda^{(k)})$, iteration is as follows

$$\begin{cases} x^{(k+1)} = \text{argmin}_x L(x, y^{(k)}, \lambda^{(k)}) \\ y^{(k+1)} = \text{argmin}_y L(x^{(k+1)}, y, \lambda^{(k)}) \\ \lambda^{(k+1)} = \lambda^{(k)} - \gamma\beta(Ax^{(k+1)} + By^{(k+1)} - b) \end{cases} \quad (10)$$

The convergence result of (10) has been established when $\gamma \in (0, (\sqrt{5}+1)/2)$ in Glowinski (1989). ADM has also recently been applied to total variation based image restoration and reconstruction (Esser, 2009).

We now investigate how ADM can be applied to solve the recovery of jointly sparse vectors in MMV model, we consider the following constrained optimization problem

$$\min_X \|X\|_{2,1}, \quad \text{s.t.} \quad \|AX - B\|_F < \delta \quad (11)$$

this is a relaxation of problem (4) by assuming that the measurement data contain noise and the linear equation $AX = B$ does not exactly hold. It is easy to see that (11) is equivalent to the following unconstrained problem

$$\min_X \{\|X\|_{2,1} + \frac{1}{2\mu}\|AX - B\|_F^2\} \quad (12)$$

When δ and μ approach to 0, the solutions of (11) and (12) approach to the solutions of (4). Thus, we focus on deriving an efficient algorithm for (12) based on the ADM. We first introduce an auxiliary matrix variable $E \in R^{m\times n}$ to measure the residue between AX and B, then (12) is equivalent to

$$\min_{X,E} \{\|X\|_{2,1} + \frac{1}{2\mu}\|E\|_F^2\} \quad \text{s.t.} \quad AX + E = B \quad (13)$$

The augmented Lagrangian function of (13) is

$$L(X, E, Y) = \|X\|_{2,1} + \frac{1}{2\mu}\|E\|_F^2 - < Y, AX + E - B > + \frac{\beta}{2}\|AX + E - B\|_F^2 \quad (14)$$

where $Y \in R^{m\times n}$ is the Lagrangian multiplier and the augmented Lagrangian subproblem is of the form

$$\min_{X,E} L(X, E, Y) \quad (15)$$

Given $(X^{(k)}, E^{(k)}, Y^{(k)})$, we will derive update rules for $(X^{(k+1)}, E^{(k+1)}, Y^{(k+1)})$ based on the ADM. First, given $(X^{(k)}, Y^{(k)})$, we update $E^{(k+1)}$. By removing terms that do not depend on E and adding proper terms not dependent on E, the optimization (15) with respect to E is equivalent to

$$\min_E \Big\{ \frac{1}{2\mu}\|E\|_F^2 - < Y^{(k)}, E + AX^{(k)} - B > + \frac{\beta}{2}\|E + AX^{(k)} - B\|_F^2 \Big\} \quad (16)$$

This is a quadratic optimization problem with respect to E, by letting the gradient of the objective function with respect to E to be 0, we have

$$(\frac{1}{\mu} + \beta)E + \beta(AX^{(k)} - B) - Y^{(k)} = 0 \quad (17)$$

it is easy to obtain that the minimizer of (16) is given by

$$E^{(k+1)} = \frac{\mu\beta}{1+\mu\beta}\big[\frac{1}{\beta}Y^{(k)} - (AX^{(k)} - B)\big] \quad (18)$$

Next, given $(X^{(k)}, E^{(k+1)}, Y^{(k)})$, consider the iteration with respect to X, by deleting and adding appropriate terms that do not depend on X and after some manipulations, the optimization problem (15) is equivalent to the following optimization problem

$$\min_X \Big\{\|X\|_{2,1} + \frac{\beta}{2}\|AX + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)}\|_F^2\Big\} \quad (19)$$

As in J.Yang et al. (2010), we do not solve (19) exactly, instead we approximate the second term in the objective function by its Taylor expansion at $x^{(k)}$ up to the second order as

$$\|AX + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)}\|_F^2 \approx \|AX^{(k)} + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)}\|_F^2 + 2 < G^{(k)}, X - X^{(k)} > + \frac{1}{\tau}\|X - X^{(k)}\|_F^2 \quad (20)$$

where $\tau > 0$ is a proximal parameter and

$$G^{(k)} = A^T(AX^{(k)} + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)}) \quad (21)$$

is the gradient of the function $\|AX + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)}\|_F^2$ evaluated at $x^{(k)}$. Drop the constant term in (20), the optimization problem (19) is approximated by the following optimization,

$$\min_X \Big\{\|X\|_{2,1} + \beta < G^{(k)}, X - X^{(k)} > + \frac{\beta}{2\tau}\|X - X^{(k)}\|_F^2\Big\} \quad (22)$$

by adding an appropriate constant term and after simple manipulations, it is easy to see that this optimization problem is equivalent to

$$\min_X \ \{\frac{\tau}{\beta}\|X\|_{2,1} + \frac{1}{2}\|X - (X^{(k)} - \tau G^{(k)})\|_F^2\} \quad (23)$$

There exists a closed-form solution to this matrix $(2,1)$-norm optimization problem. Recall that in the

classical SMV model of CS, when solving the following l_1-minimization problem

$$x^* = \arg\min_x \{\lambda\|x\|_1 + \frac{1}{2}\|x - c\|_2^2\} \tag{24}$$

where x is a vector variable and c is a constant vector, a well-known result is that the closed-form solution of (24) is given by

$$x_i^* = \text{Shrink}(c_i, \lambda) \triangleq \begin{cases} c_i - \lambda, & \text{if } c_i > \lambda \\ c_i + \lambda, & \text{if } c_i < -\lambda \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

Here, the function $\text{Shrink}(\cdot,\cdot)$ is the so-called *iterative shrinkage-thresholding* (or *soft-thresholding*) function (Donoho, 1995). Since the matrix $(2,1)$-norm $\|X\|_{2,1}$ is the l_1-norm of the vector consisting of the l_2-norm of rows of X as its components, we have the following lemma.

Lemma 3.1. Consider the following optimization problem

$$X^* = \arg\min_X \{\lambda\|X\|_{2,1} + \frac{1}{2}\|X - C\|_F^2\} \tag{26}$$

where X^*, X, C are matrices of same dimension and C is a constant matrix, the minimizer of (26) is given by

$$X^* = \text{Row\_ Shrink}(C, \lambda) \tag{27}$$

where the function $\text{Row\_ Shrink}(C, \lambda)$ is defined as follows: let the ith rows of matrices X^* and C be $(x^*)^i$ and c^i, respectively, then

$$(x^*)^i = \begin{cases} \frac{\|c^i\|_2 - \lambda}{\|c^i\|_2} c^i, & \text{if } \|c^i\|_2 > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

Proof. By the definition of the matrix $(2,1)$-norm, the original optimization problem (26) is equivalent to the following row-wise optimization problems for all rows of X and C,

$$(x^*)^i = \arg\min_{x^i} \{\lambda\|x^i\|_2 + \frac{1}{2}\|x^i - c^i\|_2^2\} \tag{29}$$

where x^i is the ith row of X. Let $f(x^i) \triangleq \lambda\|x^i\|_2 + \frac{1}{2}\|x^i - c^i\|_2^2$, then if $\|c^i\|_2 \leq \lambda$, let $x^i = \theta c^i$, θ is a scalar parameter, then $f(\theta c^i) = \lambda\|\theta c^i\|_2 + \frac{1}{2}\|(\theta - 1)c^i\|_2^2 \geq \frac{1}{2}(\theta^2 + 1)\|c^i\|_2^2$, this implies that when $\theta = 0$, i.e., $x^i = 0$, the objective function achieves its minimum. If $\|c^i\|_2 > \lambda$, let $x^i = \theta c^i$, then $f(\theta c^i) = (\lambda\theta + \frac{1}{2}(\theta - 1)^2\|c^i\|_2)\|c^i\|_2$, if the minimum of f is achieved at θ_0, θ_0 must be a stationary point. Compute the derivative of $f(\theta c^i)$ with respect θ and let it be 0, we have $\theta_0 = \frac{\|c^i\|_2 - \lambda}{\|c^i\|_2}$. This completes the proof of the Lemma.

According to Lemma 3.1, the solution to (23) is

$$X^{(k+1)} = \text{Row\_ Shrink}(X^{(k)} - \tau G^{(k)}, \frac{\tau}{\beta}) \tag{30}$$

Finally, the Lagrangian multiplier is updated by

$$Y^{(k+1)} = Y^{(k)} - \gamma\beta(AX^{(k+1)} + E^{(k+1)} - B) \tag{31}$$

In J.Yang et al. (2010), for vector l_1-norm minimization, it has been proven that if the parameters satisfy $\tau\lambda_{\max} + \gamma < 2$, where $\lambda_{\max}$ is the maximum eigenvalue of A^TA, then the iteration sequence converges to a solution of the optimization problem. The theoretical result about the convergence of our matrix iteration sequence is left as a future research topic, numerical simulations in the following section show good convergence of the proposed method.

The MMV-ADM is actually a first-order primal-dual algorithm because it iterates the primal variable X and the dual variable Y simultaneously. Furthermore, it updates the error at each iteration so as to make the algorithm converge quickly, this is a striking feature of MMV-ADM than existing methods. In summery, the overall MMV-ADM algorithm is depicted in **Algorithm 1**.

Algorithm 1 MMV-ADM algorithm

Input: Sensing matrix A, multiple measurement data matrix B, parameters μ,β,τ, and γ.

Initialization: Randomly initialize $X^{(0)}$, $Y^{(0)}$. $k = 0$;

while (*not converged*)

1. $E^{(k+1)} = \frac{\mu\beta}{1+\mu\beta}\big[\frac{1}{\beta}Y^{(k)} - (AX^{(k)} - B)\big]$.
2. $G^{(k)} = A^T(AX^{(k)} + E^{(k+1)} - B - \frac{1}{\beta}Y^{(k)})$.
3. $X^{(k+1)} = \text{Row\_ Shrink}(X^{(k)} - \tau G^{(k)}, \frac{\tau}{\beta})$.
4. $Y^{(k+1)} = Y^{(k)} - \gamma\beta(AX^{(k+1)} + E^{(k+1)} - B)$
5. $k = k + 1$;

end

Output: The jointly sparse signal X, the Lagrangian multiplier Y and the residue E.

4 EXPERIMENTS

In this section, we conduct numerical simulations to compare our MMV-ADM algorithm with the MMVprox method in L. Sun et al. (2009). As in L. Sun et al. (2009) and J. Yang et al. (2010), we use synthetic data to test the algorithms. For each simulation, we first randomly generate an $m \times d$ matrix A with its entries obeying the normal distribution $N(0,1)$ of mean 0 and variance 1, then we generate a $d \times n$ sparse matrix $\bar{X}$ with a given sparsity level, which is characterized by its number of nonzero rows

k. Given k, we first randomly generate k different numbers in $1, \ldots, d$, indicating the indices of nonzero rows in $\bar{X}$. For each nonzero row, we then randomly generate its elements following normal distribution of mean 0 and variance 1. This $\bar{X}$ is reserved as the ground truth for comparison with the recovered signal X later. Using $\bar{X}$, we generate a $m \times n$ matrix by $B = A\bar{X} + E$ where elements of E are Gaussian noise of mean 0 and standard deviation σ.

To measure the accuracy of the recovered signal X, we use the following relative error as in J.Yang et al. (2010),

$$\text{RelErr} = \frac{\|X - \bar{X}\|_F}{\|\bar{X}\|_F} \tag{32}$$

In order to make comparisons reasonable, this relative error is also used to measure the performance of the MMVprox algorithm, while in L. Sun et al. (2009), they utilized the mean squared error $\|X - \bar{X}\|_F/\sqrt{dn}$ instead. Completely fair comparisons between two algorithms are impossible because two methods use different strategies and they have different parameters, so in our simulations, we put them together into a Matlab procedure so that they have similar initial conditions and stoping conditions. We also run them on the same computer, making the CPU time comparison is reasonable.

Throughout our experiments, we fix the parameters of MMV-ADM as $\mu = 0.0001$, $\beta = 0.2$, $\tau = 0.8$, $\gamma = 0.5$, and $\varepsilon = 0.001$, where ε specifies the stoping condition $\|X^{(k+1)} - X^{(k)}\|_F / \|X^{(k)}\|_F < \varepsilon$.

A remarkable advantage of MMV-ADM over MMVprox is its fast speed. Fig. 1 shows the average relative error versus the iteration numbers of the two algorithms when the signals are noise free, i.e., $\sigma = 0$, both the horizontal and vertical axes are in log scales. The relative error is obtained by averaging 10 runs of each algorithm with random initializations. The parameters are: $m = 50$, $d = 100$, $n = 80$. As can be seen, MMV-ADM converges much faster than MMVprox, in about 50 iterations, the averaged relative error reaches as low as 10^{-5}, while it takes MMVprox several thousands of iterations to reach similar relative error.

Fig. 2 shows the average relative error versus the iteration numbers of the two algorithms when the measurement data B are contaminated by Gaussian noise with $\sigma = 0.001$. The parameters are the same as in Fig. 1. MMV-ADM also converges faster than MMVprox in less than 100 iterations with the averaged relative error 10^{-3}.

It must also be noted that with increasing iterations, MMVprox can achieve lower relative errors than MMV-ADM, and the latter has a trend that the relative error is increasing slightly (for noising signals), a phenomenon that has also been observed and explained in J.Yang et al. (2010). This is because MMV-ADM is an approximating algorithm, its main merit is its fast speed in solving a problem to an accepted accuracy. Also as pointed out there, a usually overlooked point in evaluating the performance of an algorithm is that algorithm speed should be evaluated relative to solution accuracy. J. Yang et al. also concluded that solving l_1-problem to high accuracy is not necessary whenever observed data are contaminated by noise. The value of our MMV-ADM algorithm lies in that within only 100 iterations it achieves small relative errors, while MMVprox must run several thousands of iterations to achieve similar results, although MMVprox can further reduce the relative errors with increasing iterations, it is not necessary to reach such high accuracy for noisy measurement data.

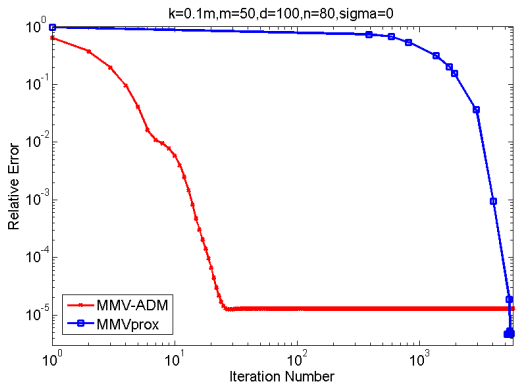

Figure 1: Relative errors versus iteration numbers for signals free of noise.

Next, we compare the performance of the two algorithms for different combinations of (m, k). Table 1 and Table 2 show the results, where "Iter", "RelErr" and "AvTi" are iteration numbers, relative errors and average CPU time, respectively. All parameters are the same as Fig. 2. One can see that MMV-ADM achieves comparable accuracy with MMVprox in about only 45-65 iterations (around 0.1 seconds), while MMVprox will take thousands of iterations (about 4 seconds).

Fig.3 and Fig.4 show the performance of two algorithms with increasing m. We fix $n = 2m$, $d = 4m$, and then increase m from 10 with stepsize 10. Fig.3 shows that when m increases from 10 to 350, the iteration number of MMV-ADM increases only from about 40

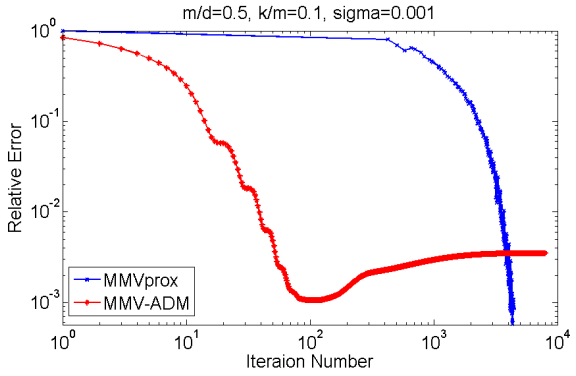

Figure 2: Relative errors versus iteration numbers for noisy signals.

Table 1: Performance of our MMV-ADM algorithm

$d = 100$		**MMV-ADM**		
m/d	k/m	Iter	RelErr	AvTi
0.5	0.1	48.7	0.00556	0.0939
0.5	0.2	62.1	0.00631	0.1242
0.4	0.1	49.1	0.00602	0.0863
0.4	0.2	63.9	0.00637	0.1168
0.3	0.1	47.4	0.00768	0.0840
0.3	0.2	65.8	0.00931	0.1188

Table 2: Performance of MMVprox algorithm

$d = 100$		**MMVprox**		
m/d	k/m	Iter	RelErr	AvTi
0.5	0.1	4072	0.0015	3.7832
0.5	0.2	4147.8	0.0018	3.8291
0.4	0.1	4179.6	0.0047	3.4934
0.4	0.2	4132.1	0.0055	3.4767
0.3	0.1	5686.3	0.0212	4.2048
0.3	0.2	5445.2	0.0062	3.9961

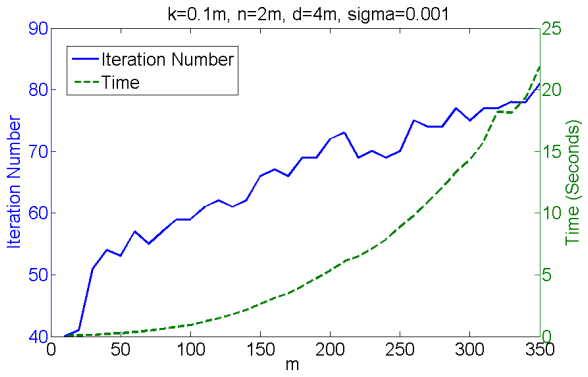

Figure 3: Iteration numbers and CPU time of MMV-ADM versus m.

to 80, this reveals the almost constant iteration number of MMV-ADM. For all cases, the algorithm stops when the relative error decreases to the order of 10^{-3}. Fig.3 also shows the CPU time of MMV-ADM which increases from about 0.1 seconds to about 23 seconds. Fig.4 shows the iteration numbers and CPU time for the MMVprox algorithm when m is increased from 10 to 200. Both are much larger than those of MMV-ADM. When m is greater than 200, the computation time grows rapidly and become unaccepted, so we have not simulated the cases when $m > 200$.

These results show that MMV-ADM scales much better to large-size problems than MMVprox. Although the computation complexity of both MMV-ADM and MMVprox is $O(mdn)$, MMV-ADM converges within dozens of iterations while MMVprox converges after thousands of iterations. This is because in ADM-MMV, the residue matrix E is updated at every iteration so that the error converges to small value quickly. In MMVprox, the projection $\mathbf{w}_{t,s} = P_{\mathbf{z}_{t-1}}(\gamma F(\mathbf{w}_{t,s-1})), \|\mathbf{w}_{t,s} - \mathbf{w}_{t,s-1}\|_2 \leq \delta$ (see L. Sun et al. (2009) for details) usually takes thousands of iterations to converge to an accepted error, although in theory it needs only two iterations to converge, searching for $\mathbf{w}_{t,s}$ close enough to $\mathbf{w}_{t,s-1}$ consumes thousands of iterations. In addition, when the problem dimension (m in our numerical experiments) increases, the iteration number increases not too much, but the total time increases significantly as indicated in Fig. 4.

Finally, Fig. 5 plots the relative error versus the iteration numbers of the MMV-ADM for $\bar{X}$ with various number k of nonzero rows, it can be seen that for different k values, the algorithm behaves similarly.

5 CONCLUSION

In this paper, we propose a fast algorithm for jointly sparse vector recovery in multiple measurement vector (MMV) model of compressive sensing. The MMV sparse signal recovery is formulated as a matrix $(2, 1)$-

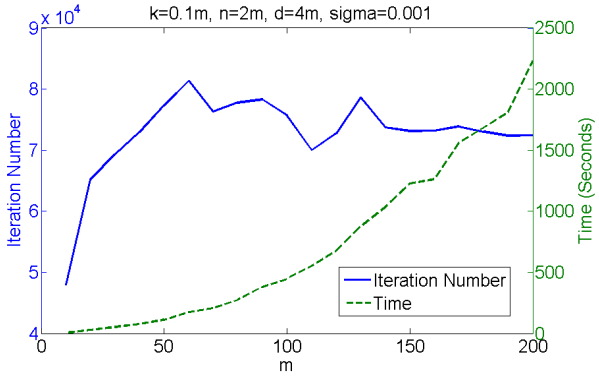

Figure 4: Iteration numbers and CPU time of MMVprox versus m.

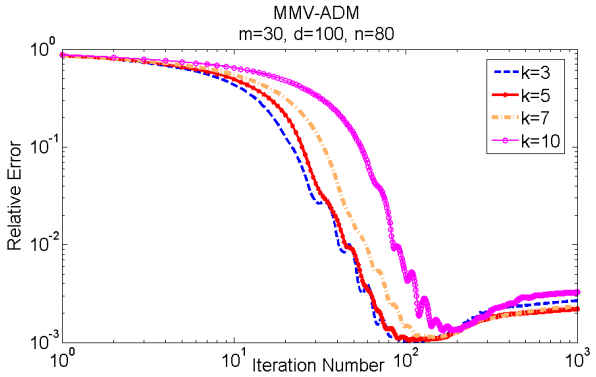

Figure 5: Relative errors of MMV-ADM versus iteration numbers for different k.

norm minimization problem with matrix equality constraint. The proposed algorithm, called MMV-ADM, is based on the alternating direction algorithm of the augmented Lagrangian multiplier method. The MMV-ADM alternately updates the signal matrix, the multiplier and the residue, and all update rules only involve matrix or vector multiplications and summations, so it is simple, easy to implement and much faster than the state-of-the-art method MMVprox. Numerical simulations show that MMV-ADM is at least dozens of times faster than MMVprox with comparable recovery accuracy. The theoretical treatment of convergence of the MMV-ADM algorithm and its applications to real problems are the future research topics.

Acknowledgements

This work is supported by the NSFC(60873133), 863(2008AA02Z310) and 973(2009CB320901) in China.

References

A. Beck and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, **2**(1):183–202.

E. J. Candes, J. Romberg, and T. Tao (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, **52**(2): 489–509.

J. Chen and X. Huo (2006). Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Transactions on Signal Processing*, **54**(12):4634–4643.

S. F. Cotter, B. D. Rao, Kjersti Engan, and K. Kreutz-Delgado (2005). Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, **53**(7):2477–2488.

D. L. Donoho (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, **41**(3):613–627.

D. L. Donoho (2006a). Compressed sensing. *IEEE Transactions on Information Theory*, **52**(4):1289–1306.

D. L. Donoho (2006b). For most large underdetermined systems of linear equations, the minimal l_1-norm solution is also the sparsest solution. *Commun. Pure Appl. Math.*, **59**(7):907–934.

D. L. Donoho and Y. Tsaig (2006). Fast solution of l_1-norm minimization problems when the solution may be sparse. preprint, http://www.stanford.edu/tsaig/research.html, 2006.

Y. C. Eldar and M. Mishali (2009). Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*,**55**(11):5302–5316.

E. Esser (2009). Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman. *CAM Report*, TR09-31, UCLA.

S. Erickson and C. Sabatti (2008). Empirical Bayes estimation of a sparse vector of gene expression changes.

Statistical Applications in Genetics and Molecular Biology, **4**(1):22.

R. Glowinski, and P. Le Tallec (1989). Augmented Lagrangian and operator splitting methods in nonlinear mechanics. *SIAM*.

I. F. Gorodnitsky and B. D. Rao (1997). Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions Signal Processing*, **45**(3):600–616.

S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky (2007). An interior point method for large-scale l_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, **1**(4):606–617.

D. Malioutov, M. Cetin, and A. Willsky (2003). Source localization by enforcing sparsity through a laplacian. In *IEEE Workshop on Statistical Signal Processing*, 553–556.

A. Nemirovski (2005). Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, **15**(1):229–251.

M. Osborne, B. Presnell, and B. Turlach (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–404.

M. Stojnic, F. Parvaresh and B. Hassibi (2009). On the reconstruction of block-sparse signals with an optimal number of measurements. *IEEE Transactions Signal Processing*, **57**(8):3075–3085.

R. T. Rockafellar (1973). The multiplier method of Hestenes and Powell applied to convex programming. *J. Optimiz. Theory App.*, **12**(6):555–562.

P. Stoica and R. Moses (1997). Introduction to Spectral Analysis. Upper Saddle River, NJ: Prentice-Hall.

J. A. Tropp, A. C. Gilbert, and M. J. Strauss (2006a). Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Proces.*, **86**(3):572-588.

J. A. Tropp (2006b). Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Signal Proces.*, **86**(3):589-602.

L. Sun, J. Liu, J. Chen, J. Ye (2009). Efficient Recovery of Jointly Sparse Vectors. In *Adv. NIPS*.

S. Wright, R. Nowak, and M. Figueiredo (2009). Sparse reconstruction by separable approximation. *IEEE Transactions Signal Processing*, **57**(7):2479–2493.

J. Yang and Y. Zhang (2010). Alternating direction algorithms for l_1-problems in compressive sensing. *SIAM Journal on Scienctific Computing*.

A. Yang, A. Ganesh, S. Sastry and Y. Ma (2010). Fast L_1-minimization algorithms and an application in robust face recognition: a review. Technical Report, http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-13.html.