THE REMBO ALGORITHM: ACCELERATED RECOVERY OF JOINTLY SPARSE VECTORS

Moshe Mishali and Yonina C. Eldar

Department of Electrical Engineering
Technion - Israel Institute of Technology, Haifa, Israel
moshiko@tx.technion.ac.il, yonina@ee.technion.ac.il

ABSTRACT

We address the problem of recovering a sparse solution of a linear under-determined system. Two variants of this problem are studied in the literature. One is the case of a sparse vector with only a few non-zero entries, and the other is of a sparse matrix with few rows non-identically zero. In either scenario, the recovery is known to be a difficult combinatorial procedure. In this paper, we develop a method that transforms the recovery of a sparse matrix into the vector formulation. Our method is exact as it allows to infer the sparse matrix from a single sparse solution vector. Once reduced to this basic form, known sub-optimal methods can be employed to approximate the solution. In order to further improve the performance, we derive a prototype algorithm, called ReMBo, that combines a boosting approach together with the reduction process. The boosting stage empirically improves the recovery rate of any given sub-optimal method. Numerical experiments demonstrate the superior performance of ReMBo-based methods in comparison with popular algorithms in terms of run time and empirical recovery rate when tested on random data.

1. INTRODUCTION

A fundamental rule of linear algebra is that a linearly independent system of equations allows for a single solution if and only if the number of equations equals the number of unknowns. Recently, it was shown that only a small set of linear equations is required to identify a unique solution vector of relatively high dimension, as long as the vector is sparse, namely it contains only a few non-zero entries [1],[2]. The sparsity structure compensates for the missing equations, however the recovery of the sparse vector becomes a difficult non-linear combinatorial problem. Interestingly, many sub-optimal efficient methods achieve a high recovery rate of the correct sparse solution when tested on a comprehensive set of random examples [1],[2],[3],[4],[5].

These fundamental results are the basis of compressed sensing, a rapid developing research area that addresses various aspects related to sparsity [1],[2]. In the basic model, referred to as *single measurement vector* (SMV), a vector $\mathbf{x}$ is compressed into a short vector $\mathbf{y}$ via a linear mapping. Under appropriate conditions the linear compression is invertible if $\mathbf{x}$ is sparse, i.e. $\mathbf{y}$ can capture the entire information about the sparse vector $\mathbf{x}$ [6]. Several sub-optimal techniques for practical recovery of $\mathbf{x}$ from $\mathbf{y}$ are developed in [1],[2],[3],[4],[7].

The SMV model was extended both theoretically and practically to sparse matrices, having only a few rows that are non-identically zero [8],[9]. This model is termed *multiple measurement vectors* (MMV) in the literature. It appears that a diverse range of applications suit a sparsity profile of this type (see [8] and the references therein). The recovery of the sparse matrix in MMV is also combinatorial and can be tackled with similar sub-optimal techniques [8],[9].

In this paper, we propose a strategy that simplifies the MMV recovery problem. Specifically, we prove that for every given MMV system there is a random construction of an SMV problem obtained by random combination of the measurements, such that the sparse solution of the SMV preserves the non-zero location set with probability one. This result relies on the observation that the non-zero location set is the crucial information to be recovered, since once found, the original sparse matrix is easily constructed by a simple linear inversion.

Our second contribution addresses the practical aspect of searching for the sparsest solution of the SMV, produced by our reduction method. To improve the recovery rate, we repeat the reduction step with different random combinations until a sparse solution is found. This stage is referred to as boosting and is based on simulations in which we noticed that applying a sub-optimal method to a specific measurement combination may yield an incorrect solution, while for another measurement mixture the correct solution can be obtained. Consequently, the additional iterations boost the overall chance of recovering the non-zero location set. We incorporate our results into a generic algorithm, which we call ReMBo (Reduce MMV and Boost). Numerical experiments demonstrate the fast run time of ReMBo-based techniques and their high recovery rate compared to other popular MMV algorithms.

The paper is organized as follows. In Section 2 we describe the SMV and MMV formulations and provide an overview of known results. The reduction to SMV is presented in Section 3. The ReMBo algorithm is introduced in Section 4 and evaluated in Section 5.

2. PROBLEM FORMULATION

Consider the following linear system

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \tag{1}$$

where $\mathbf{A}$ is a known $m \times n$ matrix with $m < n$. Our goal is to solve for the unknown matrix $\mathbf{X}$ of size $n \times d$ given the $m \times d$ measurement matrix $\mathbf{Y}$ where d is arbitrary. The system of (1) is written as $\mathbf{y} = \mathbf{A}\mathbf{x}$ when $d = 1$ in order to emphasize that in this case $\mathbf{x}, \mathbf{y}$ are vectors. Note that unless stated otherwise, the problem is defined over the complex field $\mathbb{C}$.

Solving for $\mathbf{X}$ is obviously ill-posed as (1) does not determine a unique solution for $m < n$. In particular, the number of unknowns in (1) is nd which is greater than the number of equations md. In order to allow for a unique solution, a prior on $\mathbf{X}$ must be incorporated. For example, suppose that $d = 1$ and $\mathbf{x} \in \mathcal{N}^{\perp}(\mathbf{A})$, where $\mathcal{N}^{\perp}(\mathbf{A})$ is the orthogonal space of $\mathcal{N}(\mathbf{A}) = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{0}\}$. Combining the linear prior

This work was supported in part by the Israel Science Foundation under Grant no. 1081/07 and by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715).

$\mathbf{x} \in \mathcal{N}^{\perp}(\mathbf{A})$ with (1) leads to a unique solution that can be obtained by simple algebraic manipulations.

In this paper, we consider a non-linear sparsity prior on $\mathbf{X}$. This prior restricts only the number of non-zeros entries in the solution, while their specific locations and values are not assumed. Note that, in contrast, a linear prior imposes a strict relation between the entries of the solution.

2.1 Single Measurement Vector ($d = 1$)

A vector $\mathbf{x}$ is called K-sparse if it contains no more than K non-zero entries. For a given vector $\mathbf{x}$ the support function $I(\mathbf{x}) = \{k \,|\, \mathbf{x}_k \neq 0\}$ describes the locations of the non-zero entries where $\mathbf{x}_k$ stands for the kth entry of $\mathbf{x}$. Thus, a K-sparse vector $\mathbf{x}$ conforms with a support size $|I(\mathbf{x})| \leq K$. A sufficient condition for the uniqueness of a K-sparse solution can be stated in terms of the Kruskal-rank of a matrix, which was first used in the seminal work of Kruskal [10]:

Definition 1 *The Kruskal-rank of $\mathbf{A}$, denoted $\sigma(\mathbf{A})$, is the maximal number q such that every set of q columns of $\mathbf{A}$ is linearly independent.*

Theorem 1 *If the vector $\bar{\mathbf{x}}$ is a K-sparse solution of $\mathbf{y} = \mathbf{Ax}$ and $\sigma(\mathbf{A}) \geq 2K$, then $\bar{\mathbf{x}}$ is the unique K-sparse solution of the system.*

Theorem 1 and its proof are given in [6],[9] with a slightly different notation of Spark($\mathbf{A}$) instead of the Kruskal rank. The SMV notation implies a linear system $\mathbf{y} = \mathbf{Ax}$ accompanied with a K-sparse prior on $\mathbf{x}$.

If $\bar{\mathbf{x}}$ is the unique K-sparse solution of the SMV problem $\mathbf{y} = \mathbf{Ax}$, then it is also the unique sparsest solution. Therefore, recovery of $\bar{\mathbf{x}}$ can be formulated as an optimization problem [1]:

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \;\; \text{s.t.} \;\; \mathbf{y} = \mathbf{Ax}, \tag{2}$$

where the pseudo-norm ℓ_0 counts the number of non-zero entries in $\mathbf{x}$. The program (2) is combinatorial and solvable, however, it is also known to be NP-hard. As an alternative, Donoho [1] and Candés *et. al.* [2] study the following relaxed convex program, referred to as basis pursuit:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \;\; \text{s.t.} \;\; \mathbf{y} = \mathbf{Ax}, \tag{3}$$

where $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |\mathbf{x}_i|$ is the sum over the magnitudes of the entries of $\mathbf{x}$. Basis pursuit and other polynomial-time methods were proved to produce the sparsest solution $\bar{\mathbf{x}}$ when K is small enough (compared to the uniqueness bound $\sigma(\mathbf{A})/2$ of Theorem 1) [5],[6]. These methods become sub-optimal when K exceeds the theoretical bounds, namely their solution can be different from the sparsest one. Interestingly, extensive simulations on random data show that $\bar{\mathbf{x}}$ can be recovered with overwhelming empirical probability for a broad range of K. Often exact recovery is accomplished even for $K > \sigma(\mathbf{A})/2$ as the uniqueness condition of Theorem 1 is only sufficient. Thus, although sub-optimal, these methods are of high practical interest.

2.2 Multiple Measurement Vector ($d > 1$)

Joint sparsity is a prior for (1) in the case $d > 1$. Under this prior, the non-zero entries of the matrix $\mathbf{X}$ are confined to a row subset, whose cardinality is known. Formally, a K-sparse matrix $\mathbf{X}$ has two properties: (I) each column is a K-sparse vector, and (II) $\mathbf{X}$ has no more than K rows that are non-identically zero. The support of $\mathbf{X}$ is denoted as

$$I(\mathbf{X}) = \{k \,|\, \mathbf{X}^k \neq \mathbf{0}\} = \bigcup_{i=1}^{d} I(\mathbf{X}_i), \tag{4}$$

where $\mathbf{X}^k, \mathbf{X}_i$ denote the kth row and the ith column of $\mathbf{X}$, respectively. Clearly, a jointly K-sparse matrix $\mathbf{X}$ conforms with $|I(\mathbf{X})| \leq K$. The system (1) is termed MMV in the literature when the joint sparsity prior holds. Evidently, the joint property of this prior distinguishes MMV from being a set of independent SMV systems.

The MMV model was studied both theoretically and practically. The following proposition shows that the uniqueness condition $\sigma(\mathbf{A}) \geq 2K$ of Theorem 1 can be relaxed for an MMV system.

Proposition 1 ([8],[9]) *If $\bar{\mathbf{X}}$ is a K-sparse solution matrix for (1), and $\sigma(\mathbf{A}) \geq 2K - (\text{rank}(\mathbf{Y}) - 1)$, then $\bar{\mathbf{X}}$ is the unique K-sparse solution matrix of (1).*

Similarly, it was shown that the combinatorial problem

$$\bar{\mathbf{X}} = \arg\min_{\mathbf{X}} |I(\mathbf{X})| \;\; \text{s.t.} \;\; \mathbf{Y} = \mathbf{AX}, \tag{5}$$

recovers the unique K-sparse solution matrix of an MMV system [9].

The optimization program (5) is also NP-hard. Several extensions of sub-optimal SMV methods have been extended to the MMV setting [8],[9]. These algorithms tackle the formulation (5) directly. As described in the next section, our approach is different in that we suggest to first simplify the combinatorial problem (5) by reducing it into the SMV formulation (2). No relaxation is performed during this reduction step, thus preserving the ability to recover $\bar{\mathbf{X}}$ exactly. In turn, SMV recovery methods can be deployed to approximate the sparse solution of the produced SMV system. Due to the special structure of our reduction method we can easily incorporate a boosting stage which further enhances the performance. This will be discussed in Section 4.

3. DIMENSION REDUCTION FOR MMV

3.1 Paradigm

Let $\bar{\mathbf{X}}$ be the unique K-sparse solution of (1) and assume $\sigma(\mathbf{A}) \geq 2K$. Our approach for the recovery of $\bar{\mathbf{X}}$ relies on the following two key observations:

1. Determining the unknown support $S = I(\bar{\mathbf{X}})$ is a crucial step in the sense that once S is found, $\bar{\mathbf{X}}$ is obtained exactly.
2. The set S can be identified with probability one from an SMV system constructed by random combinations of the columns of $\mathbf{Y}$.

These observations focus our attention on (2) rather than on (5). As we show in Section 5, besides the obvious impact on run time, our approach leads to recovery methods with an empirical recovery rate that is typically higher than a direct relaxation of (5).

3.2 Method

We now state each of our observations formally.

To prove the first observation, let $\mathbf{A}_S$ denote the matrix containing the subset of the columns of $\mathbf{A}$ whose indices belong to S. Since $\bar{\mathbf{X}}$ is K-sparse we have that $|S| \leq K$. In addition, since $\sigma(\mathbf{A}) \geq 2K > K$, the matrix $\mathbf{A}_S$ consists of linearly independent columns implying that

$$(\mathbf{A}_S)^{\dagger}\mathbf{A}_S = \mathbf{I}, \tag{6}$$

where $(\mathbf{A}_S)^{\dagger} = (\mathbf{A}_S^H\mathbf{A}_S)^{-1}\mathbf{A}_S^H$ is the Moore-Penrose pseudo-inverse and $\mathbf{A}_S^H$ denotes the conjugate transpose of $\mathbf{A}_S$. Using S the system of (1) can be written as

$$\mathbf{Y} = \mathbf{A}_S\mathbf{X}^S, \tag{7}$$

where $\mathbf{X}^S$ is the row subset of $\mathbf{X}$ indicated by S. Multiplying (7) by $(\mathbf{A}_S)^\dagger$ on the left gives

$$\mathbf{X}^S = (\mathbf{A}_S)^\dagger \mathbf{Y}. \tag{8}$$

In addition, it follows from the definition of the support set $I(\mathbf{X})$ that the ith row satisfies

$$\mathbf{X}^i = \mathbf{0}, \quad i \notin S. \tag{9}$$

Therefore (8)-(9) allow for exact recovery of $\bar{\mathbf{X}}$ once the finite set S is correctly recovered.

In order to construct the random SMV described in our second observation we quote the following definition from probability and measure theory [11],[12]:

Definition 2 *A probability distribution $\mathcal{P}$ is called absolutely continuous if every event of measure zero occurs with probability zero.*

A distribution is absolutely continuous if and only if it can be represented as an integral over an integrable density function [11],[12]. For example, Gaussian and uniform distributions have an explicit density function that is integrable and thus both are absolutely continuous. Conversely, discrete and other singular distributions are not absolutely continuous. The following theorem exploits this property to reduce (5) into (2):

Theorem 2 *Let $\bar{\mathbf{X}}$ be the unique K-sparse solution matrix of (1) with $\sigma(\mathbf{A}) \geq 2K$. In addition, let $\mathbf{a}$ be a random vector of length d with an absolutely continuous distribution and define the random vectors $\mathbf{y} = \mathbf{Y}\mathbf{a}$ and $\bar{\mathbf{x}} = \bar{\mathbf{X}}\mathbf{a}$. Consider the random SMV system $\mathbf{y} = \mathbf{A}\mathbf{x}$. Then:*

1. *For every realization of $\mathbf{a}$, the vector $\bar{\mathbf{x}}$ is the unique K-sparse solution of the SMV.*
2. *$I(\bar{\mathbf{x}}) = I(\bar{\mathbf{X}})$ with probability one.*

The proof of the theorem can be found in the full paper [13]. Note that an alternative reduction approach would be to treat the MMV system (1) as a set of independent SMV problems. However, since a single column of $\bar{\mathbf{X}}$ may contain zeros in some of the locations S, this approach requires to solve each of the d SMV systems. Instead, Theorem 2 implies that a single SMV problem is sufficient, thus reducing the overall complexity. In addition, when the sub-optimality of practical recovery methods is taken into account, it is evident that the aggregated sub-optimality of solving d independent SMV problems is higher than solving only one. We also note that reduction with random merging is better than every deterministic linear merging (when $\bar{\mathbf{X}}$ is deterministic unknown), since for the latter there are infinite counterexamples in which the merging process would fail to preserve the support set S. For example, a simple summation over the columns of $\mathbf{Y}$ may fail if the non-zero values in a single row of $\bar{\mathbf{X}}$ sum to zero. In contrast, Theorem 2 ensures that for *every* given MMV system and with probability one, the random reduction yields an SMV with the same non-zero location set. In essence, the random reduction maps counterexamples of zero-measure (which may occur) to zero-measure events (which are unlikely to occur).

4. THE REMBO ALGORITHM

Theorem 2 paves the way to a new class of MMV techniques based on reduction to an SMV. In this approach, the measurement matrix $\mathbf{Y}$ is first transformed into a single vector $\mathbf{y}$ by drawing a realization of $\mathbf{a}$ from some absolutely continuous distribution. Then, an SMV problem is solved by (2) or by any of its relaxations in order to find the support set S. Finally, the recovery of $\bar{\mathbf{X}}$ is carried out by inverting the matrix $\mathbf{A}_S$ as in (8)-(9).

Algorithm 1 ReMBo (Reduce MMV and Boost)

```
Input: $\mathbf{Y}, \mathbf{A}$
    Control Parameters: $K$, $\epsilon$, MaxIters, $\mathcal{S}$, $\mathcal{P}$
Output: $\hat{\mathbf{X}}$, $\hat{S}$, flag
 1: Set iter= 1
 2: Set flag=false
 3: while (iter $\leq$ MaxIters) and (flag is false) do
 4:     Draw a random vector $\mathbf{a}$ of length $d$ according to $\mathcal{P}$
 5:     $\mathbf{y} = \mathbf{Y}\mathbf{a}$
 6:     Solve $\mathbf{y} = \mathbf{A}\mathbf{x}$ using SMV technique $\mathcal{S}$ and denote the
        solution $\hat{\mathbf{x}}$
 7:     $\hat{S} = I(\hat{\mathbf{x}})$
 8:     if ($|\hat{S}| \leq K$) and ($\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2 \leq \epsilon$) then
 9:         flag=true
10:     else
11:         flag=flase
12:     end if
13:     Construct $\hat{\mathbf{X}}$ using $\hat{S}$ and (8)-(9)
14:     iter=iter+1
15: end while
16: return $\hat{\mathbf{X}}$, $\hat{S}$, flag
```

In practice, (2) is not solved directly and sub-optimal methods are used instead. To further improve the recovery ability of these algorithms we suggest to repeat the reduction process with different realizations of $\mathbf{a}$ until a sparse solution is identified. This strategy relies on the following empirical behavior which we noticed in simulations. Consider two K-sparse vectors $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$ having the same non-zero locations but with different values. Denote by $\mathcal{S}$ an SMV technique which is used to recover $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$ from the measurement vectors $\mathbf{A}\bar{\mathbf{x}}, \mathbf{A}\tilde{\mathbf{x}}$ respectively. Empirically, we observed that $\mathcal{S}$ may recover one of the vectors $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$ while failing to recover the other, even though their non-zero locations are the same. As far as we are aware, this behavior was not studied thoroughly yet in the literature. In fact, Monte-Carlo simulations that are typically conducted in the evaluation of sub-optimal techniques may imply a converse conclusion. For example, Candès *et. al.* [2] analyze the basis pursuit method (3) when $\mathbf{A}$ is a row subset of the discrete-time Fourier matrix. A footnote in the simulation section points out that the observed behavior seems to be independent of the exact distribution of which the non-zero entries are drawn from. This remark was also validated by other papers that conducted similar experiments. The conjecture that Monte-Carlo simulations are insensitive to distribution of the non-zero values appears to be true. Nevertheless, it is beneficial for a given SMV system to apply $\mathcal{S}$ on both measurement vectors $\mathbf{A}\bar{\mathbf{x}}, \mathbf{A}\tilde{\mathbf{x}}$. Once the crucial information of the non-zero locations is recovered, the final step of inverting $\mathbf{A}_S$ leads to the correct solution of both $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$.

The ReMBo algorithm, outlined in Algorithm 1, makes use of the reduction method and also capitalizes on the empirical behavior discussed above. In steps 4-7, the MMV system is reduced into an SMV and solved using a given SMV technique $\mathcal{S}$. These steps produce a sub-optimal solution $\hat{\mathbf{x}}$, which is examined in step 8. If $\hat{\mathbf{x}}$ is not sparse enough or is not well aligned with the measurements, then the reduction steps are repeated with another draw of the random vector $\mathbf{a}$. We refer to these additional iterations as the *boosting* step of the algorithm. Theorem 2 ensures that each of the different SMV systems of step 6 has a sparse solution that preserves the required support set S with probability one. The iterations improve the chances to recover S by changing the non-zero values of the sparse solutions. If the number

Table 1: Sub-Optimal Techniques

Model	Tag	Formal Description	Type
SMV	BP	Basis Pursuit, (3), [1],[2]	Convex relaxation
	OMP	Orthogonal Matching Pursuit, [8]	Greedy
MMV	M-BP(ℓ_1)	(5) with objective $\|\mathcal{R}_{\ell_1}(\mathbf{X})\|_1$, [9]	Convex relaxation
	M-BP(ℓ_∞)	(5) with objective $\|\mathcal{R}_{\ell_\infty}(\mathbf{X})\|_1$, [4]	Convex relaxation
	M-OMP	MMV version of OMP, [8]	Greedy
	ReMBo-BP	ReMBo with $\mathcal{S}$ = BP	Convex relaxation
	ReMBo-OMP	ReMBo with $\mathcal{S}$ = OMP	Greedy

of iterations exceed the pre-determined parameter MaxIters, then the algorithm is terminated. The content of the flag variable indicates whether $\hat{\mathbf{X}}$ represents a valid solution. If flag=false, then we may solve the MMV system by any other method. Note that step 8 allows to tune ReMBo to prefer either feasibility or sparsity according to user preference by selecting appropriate values for the parameters K, ϵ.

We now compare the behavior of ReMBo with standard MMV techniques in terms of computational complexity and recovery rate. Clearly, the complexity of SMV is lower due to the reduced number of unknowns. The reduction method itself is no more than one matrix multiplication which in practice is a negligible portion of the overall run time in popular techniques. Performance of different algorithms can also be evaluated by measuring the empirical recovery rate in a set of random tests [1],[2],[8],[9]. As we detail in the following section, for some parameter choices a single reduction iteration achieves an overall recovery rate that is higher than applying a direct MMV technique. For other parameter selections, a single iteration is not sufficient and boosting is required to increase the recovery rate of ReMBo beyond that of a standard MMV. The results indicate that ReMBo-based techniques are comparably fast even when boosting is employed.

5. NUMERICAL EXPERIMENTS

5.1 Setup

We choose $m = 20, n = 30, d = 5$ for the dimensions of (1). The matrix $\mathbf{A}$ contains real-valued i.i.d. Gaussian entries with zero mean and unit variance. For each $1 \leq K \leq 20$ we construct a K-sparse matrix $\mathbf{X}_K$ by first choosing a support uniformly at random among the $\binom{n}{K}$ options, and then drawing the non-zero values from the same distribution used for the entries of $\mathbf{A}$. Next, a given MMV algorithm is executed in order to recover $\mathbf{X}_K$ from the measurements $\mathbf{AX}_K$. We compute the empirical recovery rate as the average number of solutions that were recovered accurately up to machine precision from 500 repetitions. For ReMBo techniques, we choose $\mathcal{P}$ as an i.i.d. uniform distribution in $[-1, 1]^d$. We also collected run time data in order to qualitatively compare between the complexity of the tested techniques. The real-valued setup is chosen to reproduce the simulations of [8],[9]. However, the results are also valid for complex values.

To simplify the presentation of the results, Table 1 lists the techniques that are used throughout the experiments. Short labels are used to denote each of the techniques. The notation $\mathcal{R}_{\ell_p}(\mathbf{X})$ stands for a vector of length n such that its ith entry is equal to the ℓ_p norm of the ith row of $\mathbf{X}$. The MaxIters parameter of ReMBo-based techniques is noted in brackets, for example ReMBo-BP[1]. A default value of MaxIters = 5 is used if the brackets are omitted.

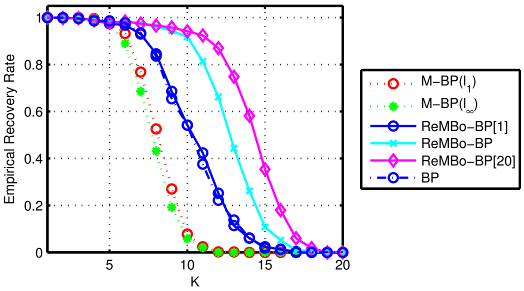


Fig. 1: Comparison of MMV algorithms based on convex relaxations. The ReMBo techniques are in solid lines. As expected, the recovery curves of ReMBo-BP[1] and BP coincide.

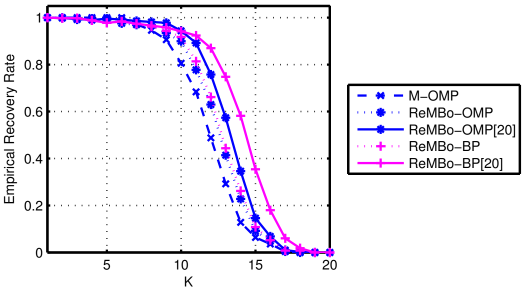
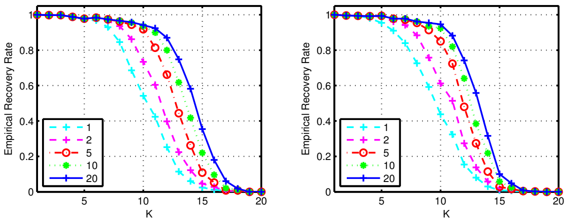


Fig. 2: A comparison between M-OMP and ReMBo-derived methods.

5.2 Results

In Fig. 1 we compare between MMV techniques based on convex relaxation of (5). For reference we used the same scale to draw the recovery rate of BP on a single measurement column. It is seen that both M-BP(ℓ_1) and M-BP(ℓ_∞) suffer from a decreased recovery rate with respect to BP. In contrast, the recovery rate of ReMBo-BP improves on BP due to the boosting effect. In addition, as revealed from Fig. 4, the average run time of ReMBo-BP is lower than the run time of both M-BP(ℓ_1) and M-BP(ℓ_∞). We point out that the M-BP techniques require the selection of a row norm. Our reduction method allows to avoid this ambiguous selection by transforming to an SMV problem.

The OMP is a variant of the greedy matching pursuit [8] and is immediately extended to M-OMP. The latter is typically faster than basis-pursuit based methods as revealed in Fig. 4. A comparison between ReMBo techniques and M-OMP is plotted in Fig. 2, demonstrating the superior per-



(a) ReMBo-BP (b) ReMBo-OMP

Fig. 3: The impact of boosting iterations for various selections of MaxIters.

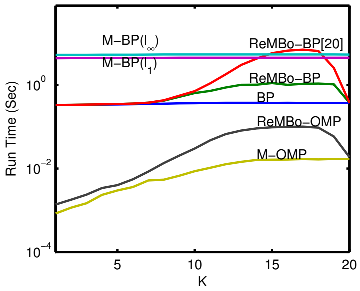
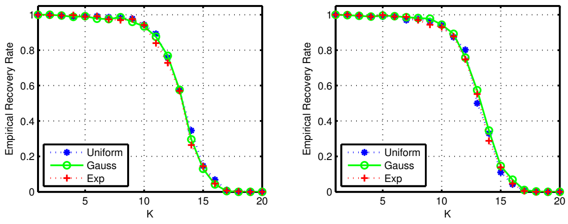


Fig. 4: Average run time of various MMV techniques.

formance of ReMBO over the entire range $1 \leq K \leq 20$. In particular, in the intermediate range $10 \leq K \leq 13$ ReMBo-OMP achieves a recovery rate that is approximately 10% higher than M-OMP. In addition, the run time of the ReMBo-OMP is close to the direct greedy approach as seen from Fig. 4.

In order to emphasize the impact of iterations, Fig. 3 depicts the recovery rate of ReMBo-BP and ReMBo-OMP for different values of MaxIters. The recovery rate at $K = 10$ is of special interest since according to Theorem 2 a value of $K \leq \sigma(\mathbf{A})/2$ is required[1] to ensure that the random instances of SMV preserves the set S. For example, a single iteration of ReMBo-BP achieves a recovery rate of 54%, while two and five iterations improve the recovery rate to 74% and 91% respectively. A higher number of iterations results in a minor improvement conforming with our default selection of MaxIters $= 5$. However, the condition of $K \leq 10$ is only sufficient and empirical recovery is allowed to some extent even for $K > 10$. In this range, repeating the reduction process more than 5 times can be beneficial. For example, ReMBo-BP[20] yields a recovery rate of 56% for $K = 14$ instead of 25% when allowing only MaxIters=5.

In Fig. 5 we compare the recovery rate when choosing the continuous distribution $\mathcal{P}$ as either uniform, Gaussian or exponential. We repeated this simulation when changing the distribution of the non-zero entries of $\mathbf{X}_K$ in the same way. The encountered behavior strengthens the conjecture that Monte-Carlo analysis is insensitive to the specific distribution of the non-zero values. Moreover, it shows that ReMBo is insensitive to the selection of $\mathcal{P}$. Note that the distribution parameters are not specified in Fig. 5 since they do not influence the results. In contrast, the boosting stage of ReMBo capitalizes on the sensitivity of common sub-optimal



(a) (b)

Fig. 5: ReMBo-OMP[20] with different selection of (a) the distribution $\mathcal{P}$, and (b) the distribution of the non-zero entries of $\mathbf{X}_K$.

methods to the specific relation between the non-zero values of the sparse solution. Clearly, Figs. 3-5 distinguish between the success recovery of a specific SMV instance and the average measure of success recovery taken over a comprehensive set of examples.

REFERENCES

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

[2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, pp. 572–588, Apr. 2006.

[4] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Process.*, vol. 86, pp. 589–602, Apr. 2006.

[5] ——, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[6] D. L. Donoho and M. Elad, "Maximal sparsity representation via $\ell 1$ minimization," *Proc. Natl. Acad. Sci.*, vol. 100, pp. 2197–2202, Mar. 2003.

[7] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A reweighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

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

[9] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Signal Processing*, vol. 54, no. 12, pp. 4634–4643, Dec. 2006.

[10] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Alg. Its Applic.*, vol. 18, no. 2, pp. 95–138, 1977.

[11] A. F. Karr, *Probability*. New York: Springer-Verlag, 1993.

[12] P. Billingsley, *Probability and Measure*. 3rd ed. New York: Wiley, 1995.

[13] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *to appear in IEEE Trans. Signal Processing*, 2008.

[1] According to [1],[2], a matrix with random entries has a full column rank and a full Kruskal rank with an overwhelming probability. In our setup the maximal value of $\sigma(\mathbf{A})$ is $m = 20$. Empirically, it was also noticed that $\text{rank}(\mathbf{Y}) = 5$ in all generated measurements.