



IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors

Moshe Mishali and Yonina C. Eldar

CCIT Report # 686
February 2008

■ ■ ■ ■ Electronics
■ ■ ■ ■ Computers
■ ■ ■ ■ Communications

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL



Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors

Moshe Mishali, *Student Member, IEEE*, and Yonina C. Eldar, *Senior Member, IEEE*

Abstract

The rapid developing area of compressed sensing suggests that a sparse vector lying in an arbitrary high dimensional space can be accurately recovered from only a small set of non-adaptive linear measurements. Under appropriate conditions on the measurement matrix, the entire information about the original sparse vector is captured in the measurements, and can be recovered using efficient polynomial methods. The vector model has been extended both theoretically and practically to a finite set of sparse vectors sharing a common non-zero location set. In this paper, we treat a broader framework in which the goal is to recover a possibly infinite set of jointly sparse vectors. Extending existing recovery methods to this model is difficult due to the infinite structure of the sparse vector set. Instead, we prove that the entire infinite set of sparse vectors can be recovered by solving a single, reduced-size finite-dimensional problem, corresponding to recovery of a finite set of sparse vectors. We then show that the problem can be further reduced to the basic recovery of a single sparse vector by randomly combining the measurement vectors. Our approach results in exact recovery of both countable and uncountable sets as it does not rely on discretization or heuristic techniques. To efficiently recover the single sparse vector produced by the last reduction step, we suggest an empirical boosting strategy that improves the recovery ability of any given sub-optimal method for recovering a sparse vector. Numerical experiments on random data demonstrate that when applied to infinite sets our strategy outperforms discretization techniques in terms of both run time and empirical recovery rate. In the finite model, our boosting algorithm is characterized by fast run time and superior recovery rate than known popular methods.

Index Terms

Basis pursuit, compressed sensing, multiple measurement vectors (MMV), orthogonal matching pursuit (OMP), sparse representation.

I. INTRODUCTION

Many signals of interest often have sparse representations, meaning that the signal is well approximated by only a few large coefficients in a specific basis. The traditional strategy to capitalize on the sparsity profile is to

first acquire the signal in a high-dimensional space, and then utilize a compression method in order to capture the dominant part of the signal in the appropriate basis. Familiar formats like MP3 (for audio signals) and JPEG (for images) implement this approach. The research area of compressed sensing (CS) challenges this strategy by suggesting that a compact representation can be acquired directly.

The fundamentals of CS were founded in the works of Donoho [1] and Candès *et. al.* [2]. In the basic model, referred to as a single measurement vector (SMV), the signal is a discrete vector \mathbf{x} of high dimension. The sensing process yields a measurement vector \mathbf{y} that is formed by inner products with a set of sensing vectors. The key observation is that \mathbf{y} can be relatively short and still contain the entire information about \mathbf{x} as long as \mathbf{x} is sparsely represented in some basis, or simply when \mathbf{x} itself contains only a few non-zero entries. An important problem in this context is whether the vector \mathbf{x} producing \mathbf{y} is unique [3]. Another well studied issue is the practical recovery of \mathbf{x} from the compressed data \mathbf{y} , which is known to be NP-hard in general. Many sub-optimal methods have been proposed for this problem [1],[2],[4],[5], which achieve a high recovery rate when tested on randomly generated sparse vectors.

The SMV model has been extended to a finite set of jointly sparse vectors having their non-zeros occurring in a common location set. The sensing vectors are applied to each of the sparse vectors resulting in multiple measurement vectors (MMV). This model is well suited for problems in Magnetoencephalography, which is a modality for imaging the brain [6],[7],[8]. It is also found in array processing [6],[9], nonparametric spectrum analysis of time series [10] and equalization of sparse communication channels [11],[12]. The issue of uniqueness in the MMV problem was addressed in [13],[14], together with extensions of SMV recovery techniques to MMV.

In this paper, we start from a broader model which consists of an infinite set of jointly sparse vectors, termed *infinite measurement vectors* (IMV). The set may be countable or uncountable (for example, when described over a continuous interval). The IMV model is broader than MMV and naturally arises in recovery problems involving analog signals, such as our earlier work on multi-band signals [15]. As we explain further in the paper, the recovery of the entire infinite set of sparse vectors in IMV models is highly complicated. A straightforward recovery approach in this context is to consider only a finite subset of vectors using discretization. However, this strategy cannot guarantee perfect recovery. Instead, we derive a reduced finite-dimensional problem from which the common non-zero location set can be inferred exactly. This paradigm relies on the observation that once the non-zero locations are identified, the original recovery problem translates into a simple linear inversion with a closed form solution.

Our first main contribution is a theoretical result showing that for every given IMV problem there is an explicit MMV counterpart with the same non-zero location set. This reduction to finite dimensions is achieved without any discretization or heuristic techniques and thus allows in principle an exact recovery of the entire set of sparse vectors. Other papers that treated problems involving infinite vector sets used discretization techniques to approximate the solution [16] or alternatively assumed an underlying discrete finite-dimensional signal model [17]. In contrast, our approach is exact as neither the IMV model nor the solution is discretized. Once the IMV problem

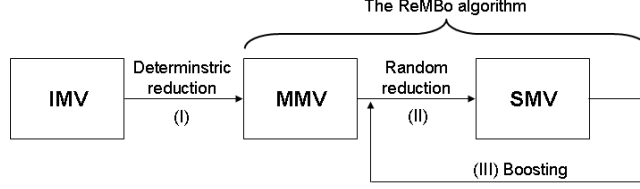


Fig. 1: The entire flow of the paper consists of: (I) a deterministic reduction from IMV to MMV, (II) a random reduction from MMV to SMV and (III) a boosting stage. The ReMBo algorithm is a formal description of the last two steps.

is reduced to an MMV problem, results developed in that context can be applied.

To further improve the recovery performance both in terms of speed and recovery rate, we develop another theoretical result allowing to identify the non-zero locations of a given MMV model from a sparse vector of a specific SMV system. As opposed to the IMV reduction, our strategy here is to construct a random SMV problem that merges the set of sparse vectors using random coefficients. We prove that this reduction preserves the crucial information of the non-zero location set with probability one.

Our final contribution treats the practical aspect of using a sub-optimal technique to find the sparse vector of an SMV problem. While examining popular SMV recovery techniques, we observed that the recovery ability depends on the exact non-zero values and not only on their locations. Based on this observation we argue that it is beneficial to draw several realizations of the merged measurement vector, by using different random combinations, until a sparse vector is identified. These iterations are referred to as the boost step of our method, since, empirically, each iteration improves the overall recovery rate of the non-zero location set. We formulate a generic algorithm, referred to as ReMBo, for the Reduction of MMV and Boosting. The ReMBo algorithm yields different recovery techniques for MMV based on the embedded SMV technique.

The results presented in this paper provide a complete flow between the recovery problem of different models. Fig. 1 depicts the entire flow which can be initiated from a given IMV system or an arbitrary MMV problem. Numerical experiments demonstrate the performance advantage of methods derived from the ReMBo algorithm over familiar MMV techniques in terms of empirical recovery rate and run time. In addition, we present a simulation emphasizing the advantage of the IMV reduction over a discretization technique.

The outline of the paper is as follows. The IMV model is introduced in Section II, where we also discuss conditions for a unique solution. The deterministic reduction method of IMV to MMV and the random reduction of MMV to SMV are developed in Sections III and IV respectively. The description of the ReMBo algorithm follows in Section V. Numerical experiments are provided in Section VI.

II. INFINITE-MEASUREMENT-VECTORS MODEL

Let \mathbf{A} be a given $m \times n$ matrix with $m < n$ and consider the parametric linear system:

$$\mathbf{y}(\lambda) = \mathbf{A}\mathbf{x}(\lambda), \quad \lambda \in \Lambda, \quad (1)$$

where Λ is some known set. Our goal is to recover the unknown vector set $\mathbf{x}(\Lambda) = \{\mathbf{x}(\lambda)\}_{\lambda \in \Lambda}$ which is referred to as the solution set, from the measurements set $\mathbf{y}(\Lambda) = \{\mathbf{y}(\lambda)\}_{\lambda \in \Lambda}$. The cardinality of the parameter set Λ is arbitrary including both finite (single or multiple element) sets and infinite sets (countable or uncountable). For example, λ can be the index of a discrete set, or alternatively a variable over a continuous interval.

Clearly, the recovery problem is not well defined unless there is a unique solution set $\mathbf{x}(\Lambda)$ for each $\mathbf{y}(\Lambda)$. However, the system of (1) does not possess a unique solution in general, since for every λ , (1) contains less equations than unknowns. Specifically, each $\mathbf{y}(\lambda)$ is a vector of length m , while the corresponding $\mathbf{x}(\lambda)$ is of length $n > m$. Therefore, in order to guarantee a unique solution an additional prior on $\mathbf{x}(\Lambda)$ must be associated with (1). Throughout this paper, we assume the joint sparsity prior, which constrains each $\mathbf{x}(\lambda)$ to have only a few non-zero entries and in addition requires that all the vectors in $\mathbf{x}(\Lambda)$ share a common non-zero location set. The system of (1) is termed IMV when Λ is infinite and the joint sparsity prior is assumed. In the sequel, this prior is formally described and is used to derive a sufficient condition for the uniqueness of the IMV solution set.

A. SMV Model

We start by describing notation and a uniqueness result for the special case of a single element set Λ , in which (1) is abbreviated as $\mathbf{y} = \mathbf{A}\mathbf{x}$. This corresponds to the well studied SMV model.

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

$$I(\mathbf{x}) = \{k \mid \mathbf{x}_k \neq 0\}, \quad (2)$$

describes the locations of the non-zero entries where \mathbf{x}_k stands for the k th 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 in this setting can be stated in terms of the Kruskal-rank of a matrix, which was first used in the seminal work of Kruskal [18]:

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 $\mathbf{y} = \mathbf{A}\mathbf{x}$ 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 [3],[14] with a slightly different notation of $\text{Spark}(\mathbf{A})$ instead of the Kruskal-rank.

TABLE I: Sparsity Models and Priors

Model	Λ Cardinality	Linear System	K -sparsity prior
SMV	1	$\mathbf{y} = \mathbf{A}\mathbf{x}$	$ I(\bar{\mathbf{x}}) \leq K$
MMV	d	$\mathbf{Y} = \mathbf{A}\mathbf{X}$	$ I(\bar{\mathbf{X}}) \leq K$
IMV	Infinite	$\mathbf{y}(\Lambda) = \mathbf{A}\mathbf{x}(\Lambda)$	$ I(\bar{\mathbf{x}}(\Lambda)) \leq K$

B. Uniqueness in IMV Models

The joint sparsity prior becomes relevant when Λ contains more than a single element. By abuse of notation, we define the support function of a vector set as the union over the support of each vector. Specifically,

$$\begin{aligned} I(\mathbf{x}(\Lambda)) &= \bigcup_{\lambda \in \Lambda} I(\mathbf{x}(\lambda)) \\ &= \{1 \leq k \leq n \mid \mathbf{x}_k(\lambda_0) \neq 0, \text{ for some } \lambda_0 \in \Lambda\}. \end{aligned} \quad (3)$$

For brevity, a jointly sparse solution set $\mathbf{x}(\Lambda)$ with $|I(\mathbf{x}(\Lambda))| \leq K$ is also called K -sparse. A K -sparse vector set $\mathbf{x}(\Lambda)$ implies two properties: (I) Each $\mathbf{x}(\lambda)$ is a K -sparse vector, and (II) the non-zero entries of $\mathbf{x}(\lambda)$ are confined to a fixed location set of size no more than K . The system of (1) is called MMV in the literature when the joint sparsity prior holds over a finite set of sparse vectors [13],[14]. Similarly, we refer to the system of (1) as IMV when Λ is an infinite set and $\bar{\mathbf{x}}(\Lambda)$ is jointly sparse. Table I summarizes the models derived from (1) for different cardinalities of the set Λ . The abbreviations used for the linear systems of MMV and IMV are clear from the context. Evidently, the joint sparsity prior is what distinguishes MMV and IMV models from being a set of independent SMV systems.

The first property of the joint sparsity prior implies that $\sigma(\mathbf{A}) \geq 2K$ is sufficient to guarantee the uniqueness of a K -sparse solution set $\bar{\mathbf{x}}(\Lambda)$, since we can consider the SMV problem $\mathbf{y}(\lambda) = \mathbf{A}\mathbf{x}(\lambda)$ for each λ separately. Exploiting the joint sparsity, we expect that a value of $\sigma(\mathbf{A})$ less than $2K$ would suffice to ensure uniqueness. Extending uniqueness results regarding MMV [13],[14] leads to the following proposition:

Proposition 1: If $\bar{\mathbf{x}}(\Lambda)$ is a K -sparse solution set for (1), and

$$\sigma(\mathbf{A}) \geq 2K - (\dim(\text{span}(\mathbf{y}(\Lambda))) - 1), \quad (4)$$

then $\bar{\mathbf{x}}(\Lambda)$ is the unique K -sparse solution set of (1).

The notation $\text{span}(\mathbf{y}(\Lambda))$ is used for the subspace of minimal dimension containing the entire vector set $\mathbf{y}(\Lambda)$. Note that $\text{span}(\mathbf{y}(\Lambda))$ is guaranteed to have finite dimension since $\mathbf{y}(\lambda)$ has finite length. For jointly sparse solution sets, Proposition 1 indeed claims that the required Kruskal-rank of \mathbf{A} can be generally lower than $2K$ of Theorem 1.

Proof: The solution set $\bar{\mathbf{x}}(\Lambda)$ is K -sparse which implies that $\dim(\text{span}(\bar{\mathbf{x}}(\Lambda))) \leq K$. It follows from the linear system of (1) that the dimension of the subspace $\text{span}(\mathbf{y}(\Lambda))$ cannot be higher than $\text{span}(\bar{\mathbf{x}}(\Lambda))$, i.e.

$r = \dim\{\text{span}(\mathbf{y}(\Lambda))\} \leq K$. From (4) we get that $\sigma(\mathbf{A}) \geq K$. Consequently, for each $\mathbf{y}(\lambda) = \mathbf{0}$ the corresponding unique K -sparse vector is $\mathbf{x}(\lambda) = \mathbf{0}$, as the null space of \mathbf{A} cannot contain other K -sparse vectors. Therefore, without loss of generality we can prove the claim for a measurement set $\mathbf{y}(\Lambda)$ with $r \geq 1$ which does not contain zero vectors.

For $r \geq 1$ there exists a finite set $\tilde{\Lambda} = \{\lambda_i\}_{i=1}^r \subseteq \Lambda$ such that the vector set $\mathbf{y}(\tilde{\Lambda})$ is linearly independent. Since $\tilde{\Lambda}$ is a finite set, $\mathbf{y}(\tilde{\Lambda}) = \mathbf{A}\mathbf{x}(\tilde{\Lambda})$ is an MMV system. According to [13],[14], the corresponding solution set $\bar{\mathbf{x}}(\tilde{\Lambda})$ is unique under the condition (4). Since $\mathbf{y}(\Lambda)$ does not contain zero vectors, every vector $\mathbf{y}(\lambda)$ belongs to some subset of r linearly independent vectors. The argument above implies the uniqueness of the corresponding subset of $\bar{\mathbf{x}}(\Lambda)$, and consequently the uniqueness of the entire solution set. ■

Note that (1) can be viewed as a sampling process, where $\bar{\mathbf{x}}(\Lambda)$ is the signal, \mathbf{A} the sampling operator and $\mathbf{y}(\Lambda)$ is the generated set of samples. In this context, the design of the sampling operator requires to determine the number of rows in \mathbf{A} such that the samples match a unique signal. However, (4) cannot be used for this task since the value of $\dim(\text{span}(\mathbf{y}(\Lambda)))$ is not known a-priori. In other words, if a matrix \mathbf{A} needs to be designed such that uniqueness is guaranteed to every K -sparse solution set, including those with $\dim\{\text{span}(\mathbf{y}(\Lambda))\} = 1$, then the condition (4) is reduced to $\sigma(\mathbf{A}) \leq 2K$ of Theorem 1.

In the remainder of the paper, we assume that a unique solution of (1) is guaranteed by either Theorem 1 or Proposition 1. In the next sections, we develop the main contributions of our work which address the recovery of the unique K -sparse solution set $\bar{\mathbf{x}}(\Lambda)$.

III. DIMENSION REDUCTION FOR INFINITE Λ

A. Optimization Viewpoint

Before discussing the IMV model we review the optimization viewpoint for the SMV and MMV problems.

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

$$\bar{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (5)$$

where the pseudo-norm ℓ_0 counts the number of non-zero entries in \mathbf{x} . In our notation the objective can be replaced by $|I(\mathbf{x})|$. Since (5) is known to be NP-hard [1],[2], several alternatives have been proposed in the literature. Donoho [1] and Candès *et. al.* [2] rigorously analyze the basis pursuit technique which uses the ℓ_1 norm instead of the ℓ_0 in (5) resulting in a tractable convex program. Various greedy techniques to approximate the sparsest solution have also been studied thoroughly [4],[5],[6]. Empirically, all these methods show a high recovery rate of the unique sparsest solution when tested on random data. Analogously, it was shown that the combinatorial problem

$$\bar{\mathbf{X}} = \arg \min_{\mathbf{X}} |I(\mathbf{X})| \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{A}\mathbf{X}, \quad (6)$$

recovers the unique K -sparse solution matrix of an MMV system [14]. This optimization problem is also NP-hard and can be tackled with similar efficient sub-optimal techniques [13],[14].

Extending the optimization viewpoint to the IMV model leads to the equivalent problem:

$$\begin{aligned} \bar{\mathbf{x}}(\Lambda) &= \arg \min_{\mathbf{x}(\Lambda)} |I(\mathbf{x}(\Lambda))| \\ \text{s.t. } \mathbf{y}(\lambda) &= \mathbf{A}\mathbf{x}(\lambda), \forall \lambda \in \Lambda. \end{aligned} \quad (7)$$

Note that in (7) there are infinitely many unknowns $\mathbf{x}(\Lambda)$, and infinitely many equations. In contrast to the finite formulation of both (5) and (6), a program of the type (7) was not analyzed in the optimization literature. The most relevant programming structures are semi-infinite programming [19] and generalized semi-infinite programming [20]. However, these formulations allow only for infinite constraints while the optimization variable is finite. This inherent intricacy of (7) remains even if the objective is relaxed by known strategies. To overcome this difficulty, we suggest to transform (7) into one of the forms known in the literature. Specifically, we show that the joint sparsity prior allows to convert (7) into a problem of the form (6), in which both the variable and the constraint set are finite.

A straightforward approach to reduce (7) to a finite-dimensional problem is to choose a finite grid $\tilde{\Lambda} \subset \Lambda$, and then solve only for $\bar{\mathbf{x}}(\tilde{\Lambda})$. This yields an MMV system corresponding to the optimization problem (6). In turn, this program can be relaxed by any of the known CS techniques. The final step is to approximate $\mathbf{x}(\Lambda)$ by interpolating the partial solution set $\bar{\mathbf{x}}(\tilde{\Lambda})$. However, a discretization approach typically results in an approximation $\mathbf{x}(\Lambda)$ that is different from the unique solution $\bar{\mathbf{x}}(\Lambda)$. Moreover, $\mathbf{x}(\Lambda)$ typically does not satisfy (1) between the grid points, that is for $\lambda \notin \tilde{\Lambda}$. This drawback of discretization happens even if a brute-force method is used to optimally find the solution set $\bar{\mathbf{x}}(\tilde{\Lambda})$ on the grid $\tilde{\Lambda}$. Furthermore, the density of the grid directly impacts the complexity of discretization techniques. For these reasons, we avoid discretization and instead propose an exact method that transforms the infinite structure of (7) into a single MMV system without losing any information. A numerical experiment illustrating the difference between our exact method and discretization is provided in Section VI-C.

B. Paradigm

In order to solve (7) exactly we split the problem into two sub-problems. One is aimed at finding the support set $S = I(\bar{\mathbf{x}}(\Lambda))$. The other reconstructs $\bar{\mathbf{x}}(\Lambda)$ from the data $\mathbf{y}(\Lambda)$ and the knowledge of S . The reason for this separation is that once S is known the linear relation of (1) can be inverted exactly. To see this, let \mathbf{A}_S denote the matrix containing the subset of the columns of \mathbf{A} whose indices belong to S . Since the solution set $\bar{\mathbf{x}}(\Lambda)$ is K -sparse we have that $|S| \leq K$. In addition, from Proposition 1, $\sigma(\mathbf{A}) \geq K$. Therefore \mathbf{A}_S consists of linearly independent columns implying that

$$(\mathbf{A}_S)^\dagger \mathbf{A}_S = \mathbf{I}, \quad (8)$$

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

$$\mathbf{y}(\lambda) = \mathbf{A}_S \mathbf{x}^S(\lambda), \quad \lambda \in \Lambda, \quad (9)$$

where the superscript $\mathbf{x}^S(\lambda)$ is the vector that consists of the entries of $\mathbf{x}(\lambda)$ in the locations S . Multiplying (9) by $(\mathbf{A}_S)^\dagger$ from both sides gives

$$\mathbf{x}^S(\lambda) = (\mathbf{A}_S)^\dagger \mathbf{y}(\lambda), \quad \lambda \in \Lambda. \quad (10)$$

In addition, it follows from the definition of the support set $I(\mathbf{x}(\Lambda))$ that

$$\mathbf{x}_i(\lambda) = 0, \quad \forall i \notin S, \lambda \in \Lambda. \quad (11)$$

Therefore (10)-(11) allow for exact recovery of $\bar{\mathbf{x}}(\Lambda)$ once the finite set S is correctly recovered.

C. Method

The essential part of our method is the first sub-problem that recovers S from the measurement set $\mathbf{y}(\Lambda)$. Our key observation is that every collection of vectors spanning the subspace $\text{span}(\mathbf{y}(\Lambda))$ contains sufficient information to recover S exactly, as incorporated in the following theorem:

Theorem 2: Suppose (1) has a unique K -sparse solution set $\bar{\mathbf{x}}(\Lambda)$ with $S = I(\bar{\mathbf{x}}(\Lambda))$ and that the matrix $\mathbf{A}_{m \times n}$ satisfies (4). Let \mathbf{V} be a matrix of m rows such that the column span of \mathbf{V} is equal to $\text{span}(\mathbf{y}(\Lambda))$. Then, the linear system

$$\mathbf{V} = \mathbf{A}\mathbf{U} \quad (12)$$

has a unique K -sparse solution $\bar{\mathbf{U}}$ and $I(\bar{\mathbf{U}}) = S$.

Proof: Let $r = \dim(\text{span}(\mathbf{y}(\Lambda)))$ and construct an $m \times r$ matrix \mathbf{Y} by taking some set of r linearly independent vectors from $\mathbf{y}(\Lambda)$. Similarly, construct the matrix $\bar{\mathbf{X}}$ of size $n \times r$ by taking the corresponding r vectors from $\bar{\mathbf{x}}(\Lambda)$. The proof is based on observing the linear system

$$\mathbf{Y} = \mathbf{A}\bar{\mathbf{X}}. \quad (13)$$

We first prove that $\bar{\mathbf{X}}$ is the unique K -sparse solution matrix of (13) and that $I(\bar{\mathbf{X}}) = S$. Based on this result, the matrix $\bar{\mathbf{U}}$ is constructed, proving the theorem.

It is easy to see that $I(\bar{\mathbf{X}}) \subseteq S$, since the columns of $\bar{\mathbf{X}}$ are a subset of $\bar{\mathbf{x}}(\Lambda)$. This means that $\bar{\mathbf{X}}$ is a K -sparse solution set of (13). Moreover, $\bar{\mathbf{X}}$ is also the unique K -sparse solution of (13) according to Proposition 1. To conclude the claim on $\bar{\mathbf{X}}$ it remains to prove that $k \in S$ implies $k \in I(\bar{\mathbf{X}})$ as the opposite direction was already proved. If $k \in S$, then for some $\lambda_0 \in \Lambda$ the k th entry of the vector $\mathbf{x}(\lambda_0)$ is non-zero. Now, if $\mathbf{x}(\lambda_0)$ is one of the columns of $\bar{\mathbf{X}}$, then the claim follows trivially. Therefore, assume that $\mathbf{x}(\lambda_0)$ is not a column of $\bar{\mathbf{X}}$. We next exploit the following lemma:

Lemma 1 ([15]): For every two matrices \mathbf{A}, \mathbf{P} , if $|I(\mathbf{P})| \leq \sigma(\mathbf{A})$ then $\text{rank}(\mathbf{P}) = \text{rank}(\mathbf{A}\mathbf{P})$.

Clearly, Lemma 1 ensures that $\text{rank}(\bar{\mathbf{X}}) = r$. In addition it follows from the same lemma that $\dim\{\text{span}(\bar{\mathbf{x}}(\Lambda))\} = r$. Thus, $\mathbf{x}(\lambda_0)$ must be a (non-trivial) linear combination of the columns of $\bar{\mathbf{X}}$. Since the k th entry of $\mathbf{x}(\lambda_0)$ is non-zero, it implies that at least one column of $\bar{\mathbf{X}}$ has a non-zero value in its k th entry, which means $k \in I(\bar{\mathbf{X}})$.

Summarizing the first step of the proof, we have that every r linearly independent columns of $\mathbf{y}(\Lambda)$ form an MMV model (13) having a unique K -sparse solution matrix $\bar{\mathbf{X}}$, such that $I(\bar{\mathbf{X}}) = S$. As the column span of \mathbf{V} is equal to the column span of \mathbf{Y} we have that $\text{rank}(\mathbf{V}) = r$. Since \mathbf{V} and \mathbf{Y} have the same rank, and \mathbf{Y} also has full column rank, we get that $\mathbf{V} = \mathbf{Y}\mathbf{R}$ for a unique matrix \mathbf{R} of r linearly independent rows. This immediately implies that $\bar{\mathbf{U}} = \bar{\mathbf{X}}\mathbf{R}$ is a solution matrix for (12). Moreover, $\bar{\mathbf{U}}$ is K -sparse, as each of its columns is a linear combination of the columns of $\bar{\mathbf{X}}$. Proposition 1 implies the uniqueness of $\bar{\mathbf{U}}$ among the K -sparse solution matrices of (12).

It remains to prove that $I(\bar{\mathbf{U}}) = I(\bar{\mathbf{X}})$. To simplify notation, we write $\bar{\mathbf{X}}^i$ for the i th row of $\bar{\mathbf{X}}$. Now, $\bar{\mathbf{U}}^i = \bar{\mathbf{X}}^i\mathbf{R}$, for every $1 \leq i \leq n$. Thus, if $\bar{\mathbf{X}}^i$ is a zero row, then so is $\bar{\mathbf{U}}^i$. However, for a non-zero row $\bar{\mathbf{X}}^i$, the corresponding row $\bar{\mathbf{U}}^i$ cannot be zero since the rows of \mathbf{R} are linearly independent. ■

The advantage of Theorem 2 is that it allows to avoid the infinite structure of (7) and to concentrate on finding the finite set S by solving the single MMV system of (12). The additional requirement of Theorem 2 is to construct a matrix \mathbf{V} having a column span equal to $\text{span}(\mathbf{y}(\Lambda))$ (i.e. the columns of \mathbf{V} are a frame for $\text{span}(\mathbf{y}(\Lambda))$). The following proposition suggests a procedure for creating a matrix \mathbf{V} with this property.

Proposition 2: If the integral

$$\mathbf{Q} = \int_{\lambda \in \Lambda} \mathbf{y}(\lambda) \mathbf{y}^H(\lambda) d\lambda, \quad (14)$$

exists, then every matrix \mathbf{V} satisfying $\mathbf{Q} = \mathbf{V}\mathbf{V}^H$ has a column span equal to $\text{span}(\mathbf{y}(\Lambda))$.

The existence of the integral in (14) translates into a finite energy requirement. Specifically, for countable Λ the integral exists if the sequence $\{\mathbf{y}_k(\lambda_i)\}_{i=1}^{\infty}$ is energy bounded in the ℓ_2 sense for every $1 \leq k \leq m$. For uncountable Λ , $\mathbf{y}_k(\lambda)$ can be viewed as a function of λ which is required to be integrable and of bounded energy in the L_2 sense for every $1 \leq k \leq m$. Note that the matrix \mathbf{Q} of (14) is positive semi-definite and thus a decomposition $\mathbf{Q} = \mathbf{V}\mathbf{V}^H$ always exists. In particular, the columns of \mathbf{V} can be chosen as the eigenvectors of \mathbf{Q} multiplied by the square-root of the corresponding eigenvalues.

Proof: For finite Λ the claim follows immediately from the fact that every two matrices \mathbf{M}, \mathbf{N} with $\mathbf{M}\mathbf{M}^H = \mathbf{N}\mathbf{N}^H$ have the same column space. Therefore, it remains to extend this property to infinite Λ .

Let $r = \dim(\text{span}(\mathbf{y}(\Lambda)))$ and define a matrix $\mathbf{Y}_{m \times r}$ as in the proof of Theorem 2. The columns of \mathbf{Y} are linearly independent and thus \mathbf{Y}^\dagger is well defined. Define the vector set $\mathbf{d}(\lambda) = \mathbf{Y}^\dagger(\mathbf{y}(\lambda))$, $\lambda \in \Lambda$, where each $\mathbf{d}(\lambda)$ is a vector of length r . By construction, the integral

$$\int_{\lambda \in \Lambda} \mathbf{d}(\lambda) \mathbf{d}^H(\lambda) d\lambda = \mathbf{Y}^\dagger \mathbf{Q} (\mathbf{Y}^\dagger)^H = \mathbf{D}\mathbf{D}^H \quad (15)$$

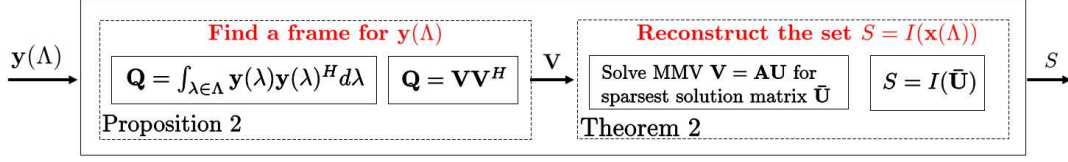


Fig. 2: The fundamental stages for the recovery of the non-zero location set S using only one finite-dimensional problem.

exists. The last equality in (15) is due to the positive semi-definiteness of the integrand. Substituting into (14) we have that $\mathbf{V}\mathbf{V}^H = (\mathbf{Y}\mathbf{D})(\mathbf{Y}\mathbf{D})^H$ which implies that the column spans of \mathbf{V} and $(\mathbf{Y}\mathbf{D})$ are the same. Since the column span of \mathbf{Y} equals to $\text{span}(\mathbf{y}(\Lambda))$, $\mathbf{d}(\Lambda)$ contains the columns of the identity matrix of size $r \times r$, and thus \mathbf{D} is invertible. In turn, this implies that $\text{span}(\mathbf{Y}) = \text{span}(\mathbf{Y}\mathbf{D})$. ■

The computation of the matrix \mathbf{Q} depends on the underlying application. In [15] we considered this approach for the reconstruction of an analog multi-band signal from point-wise samples. This class of signals are sparsely represented in the frequency domain as their Fourier transform is restricted to several disjoint intervals. Imposing a blind constraint, namely that both sampling and reconstruction are carried out without knowledge of the band locations, yields an IMV system that depends on a continuous frequency parameter. As described in [15], in this application \mathbf{Q} can be computed by evaluating correlations between the sampling sequences in the time domain. The existence of the integral in (14) corresponds to the basic requirement that the point-wise sampling process produces bounded energy sampling sequences.

Fig. 2 summarizes the reduction steps that follow from Theorem 2 and Proposition 2. The flow of Fig. 2 was first presented and proved in our earlier work [15]. The version we provide here has several improvements over the preliminary one of [15]. First, the flow is now divided into two independent logical stages and the purpose of each step is highlighted. Second, each stage has a stand-alone proof as opposed to the technique used in [15] to prove the entire scheme at once. Mathematically, this separation allows us to remove the restriction imposed in [15] on \mathbf{V} to have only orthogonal columns. Moreover, each block can be replaced by another set of operations having an equivalent functionality. In particular, the computation of the matrix \mathbf{Q} of Proposition 2 can be avoided if other methods are employed for the construction of a frame \mathbf{V} for $\text{span}(\mathbf{y}(\Lambda))$.

IV. DIMENSION REDUCTION FOR FINITE Λ

A. Objective

We now address the finite case of an MMV system

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \quad (16)$$

with \mathbf{A} an $m \times n$ rectangular matrix as before. Following the convention of Table I, \mathbf{Y} is an $m \times d$ matrix, and the dimensions of \mathbf{X} are $n \times d$. We assume that a unique K -sparse solution matrix $\bar{\mathbf{X}}$ with no more than K

non-identical zero rows exists. The unique solution $\bar{\mathbf{X}}$ can be found by the optimization problem (6), which has known relaxations to tractable techniques. Our goal in this section is to rely on ideas developed in the context of the IMV model in order to reduce the dimension of the optimization variable of (6) before performing any relaxation. Note that the MMV system (16) is arbitrary and the results developed in the sequel do not assume a preceding stage of reduction from IMV.

Applying the same paradigm of the infinite scenario, we aim to recover the support set $S = I(\bar{\mathbf{X}})$. This set contains the crucial information in the sense that once S is recovered the solution is obtained by (10)-(11), namely by inverting the relevant columns of \mathbf{A} . An immediate corollary of Theorem 2 is that if \mathbf{Y} does not have full column rank, then (16) can be reduced by taking an appropriate column subset of \mathbf{Y} . However, we wish to improve on this trivial result. Specifically, we intend to find the support set S from a single SMV optimization of the form (5). Such a reduction method is beneficial as the dimensions of the unknown variable in (5) is n while in (6) it is nd .

B. Method

Our approach is to randomly merge the columns of \mathbf{Y} into a single vector \mathbf{y} . We then show that the set S can be extracted from the random SMV $\mathbf{y} = \mathbf{A}\mathbf{x}$. In order to derive this result rigorously we rely on the following definition from probability and measure theory [21],[22]:

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 [21],[22]. 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 (6) into (5):

Theorem 3: Let $\bar{\mathbf{X}}$ be the unique K -sparse solution matrix of (16) 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 $\bar{\mathbf{x}} = \bar{\mathbf{X}}\mathbf{a}$. Then, for the random SMV system $\mathbf{y} = \mathbf{A}\mathbf{x}$ we have:

- 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.

Proof: For every realization of \mathbf{a} , the vector $\bar{\mathbf{x}}$ is a linear combination of the jointly K -sparse columns of $\bar{\mathbf{X}}$, and thus $\bar{\mathbf{x}}$ is K -sparse. It is easy to see that $\bar{\mathbf{x}}$ satisfies the SMV system and that Theorem 1 implies its uniqueness among the K -sparse vectors.

Denote $S = I(\bar{\mathbf{X}})$ and observe that the previous argument implies that $I(\bar{\mathbf{x}}) \subseteq S$ for every realization of \mathbf{a} . Therefore, it remains to prove that the event $I(\bar{\mathbf{x}}) = S$ occurs with probability one. Expressing this event in terms

of the rows of $\bar{\mathbf{X}}$ gives

$$\begin{aligned} \text{Prob}\{I(\bar{\mathbf{x}}) = S\} &= \text{Prob}\{\mathbf{a} \notin \mathcal{N}(\bar{\mathbf{X}}^i), \quad \forall i \in S\} \\ &= 1 - \text{Prob}\left\{\mathbf{a} \in \bigcup_{i \in S} \mathcal{N}(\bar{\mathbf{X}}^i)\right\}, \end{aligned} \quad (17)$$

where $\bar{\mathbf{X}}^i$ denotes the i th row of $\bar{\mathbf{X}}$, and $\mathcal{N}(\bar{\mathbf{X}}^i) = \{\mathbf{v} \mid \bar{\mathbf{X}}^i \mathbf{v} = \mathbf{0}\}$ is the null space of that row. Now, for every $i \in S$, the row $\bar{\mathbf{X}}^i$ is not identically zero, and thus the dimension of $\mathcal{N}(\bar{\mathbf{X}}^i)$ is $d - 1$. In other words, for every $i \in S$, we have that $\mathcal{N}(\bar{\mathbf{X}}^i)$ has a zero measure in the underlying sample space of \mathbf{a} , which can be either \mathbb{R}^d or \mathbb{C}^d . The union in (17) is over the finite set S and thus has also a measure zero. The absolute continuity of the distribution of \mathbf{a} concludes the proof. \blacksquare

The randomness of \mathbf{a} plays a main role in the reduction method suggested by Theorem 3. In fact, random merging is a best choice in the sense that for every deterministic linear merging 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 3 ensures that for every given MMV system and with probability one, the random reduction yields an SMV with the same non-zero location set.

The result of Theorem 3 resembles a result of [23], in which the authors suggested merging the columns of \mathbf{Y} using an ordinary summation. The non-zero locations were then estimated using a one step greedy algorithm (OSGA). It was shown that if the entries of $\bar{\mathbf{X}}$ are random, drawn independently from a Gaussian distribution, then the set S can be recovered by OSGA with probability approaching one as long as $d \rightarrow \infty$, that is when the number of columns in each of the matrices $\mathbf{Y}, \bar{\mathbf{X}}$ is taken to infinity. In contrast, our method does not assume a stochastic prior on the solution set $\bar{\mathbf{X}}$. Moreover, Theorem 3 holds with probability one for arbitrary finite and fixed values of d .

V. THE REMBO ALGORITHM

Theorem 3 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 of the type (5) is solved 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 (10)-(11).

Since (5) is NP-hard it is not solved explicitly in practice. Instead, many efficient sub-optimal techniques have been proposed in the literature that are designed to be tractable but no longer guarantee a recovery of the unique sparsest solution. Interestingly, we have discovered that repeating the reduction process of the previous section with different realizations of \mathbf{a} is advantageous due to the following empirical behavior of these sub-optimal techniques. 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 CS techniques may imply a converse conclusion. For example, Candès *et. al.* [2] analyzed the basis pursuit method for SMV 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 $\mathbf{A}_{\mathcal{S}}$ 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 term these additional iterations the boosting step of the algorithm. Theorem 3 ensures that each of the different SMV systems of step 6 has a sparse solution that preserves the required support set \mathcal{S} with probability one. The iterations improve the chances to recover \mathcal{S} by changing the non-zero values of the sparse solutions. Note that if the number 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.

In general, CS techniques can be clustered into two groups. Those of the first group search for the sparsest feasible solution. The other group contains approximation methods that fix the sparsity to a user-defined value and determine a solution in this set that is best aligned with the data. For example, basis pursuit [24] belongs to the first group, while matching pursuit [25] with a fixed number of iterations belongs to the second group. The ReMBo algorithm can be tuned to prefer either feasibility or sparsity according to user preference by selecting appropriate values for the parameters K, ϵ . However, it is recommended to avoid an approximation technique of the second group when constraining only K to a desired value. The reason is that such a method makes the condition of step 8 always true, and thus no boosting will occur.

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 typical CS techniques. Performance of different algorithms can also be evaluated by measuring the empirical recovery rate in a set of random tests [1],[2],[13],[14]. As we detail in the following section, for some parameters 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

Algorithm 1 ReMBo (Reduce MMV and Boost)

Input: \mathbf{Y}, \mathbf{A}
Control Parameters: $K, \epsilon, \text{MaxIters}, \mathcal{S}, \mathcal{P}$
Output: $\hat{\mathbf{X}}, \hat{\mathcal{S}}, \text{flag}$

```

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  $\mathcal{S}$ . Denote the solution  $\hat{\mathbf{x}}$ .
7:    $\hat{\mathcal{S}} = I(\hat{\mathbf{x}})$ 
8:   if ( $|\hat{\mathcal{S}}| \leq K$ ) and ( $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2 \leq \epsilon$ ) then
9:     flag=true
10:  else
11:    flag=false
12:  end if
13:  Construct  $\hat{\mathbf{X}}$  using  $\hat{\mathcal{S}}$  and (10)-(11)
14:  iter=iter+1
15: end while
16: return  $\hat{\mathbf{X}}, \hat{\mathcal{S}}, \text{flag}$ 

```

to increase the recovery rate of a ReMBo technique beyond that of a standard MMV. The results indicate that ReMBo based techniques are comparably fast even when boosting is employed.

VI. NUMERICAL EXPERIMENTS

In this section we begin by evaluating the reduction and boosting approach for MMV systems. The behavior of the ReMBo algorithm is demonstrated when the produced SMV is solved using a sub-optimal method. Two representative MMV techniques are derived from Algorithm 1 and compared with other popular MMV techniques. We then present an experiment that demonstrates the benefits of the IMV reduction flow over a discretization technique.

A. Evaluating ReMBo

We choose $m = 20, n = 30, d = 5$ for the dimensions of (16). The following steps are repeated 500 times for each MMV technique:

- 1) A real-valued matrix \mathbf{A} of size 20×30 is constructed by drawing each of its entries independently from a Gaussian distribution with zero mean and variance one.
- 2) For each value of $1 \leq K \leq 20$ we construct a K -sparse real-valued solution matrix $\bar{\mathbf{X}}_K$ of size 30×5 . The non-zero values of $\bar{\mathbf{X}}_K$ are also drawn from a Gaussian distribution in the same way described before.
- 3) The MMV technique that is being tested is executed in order to recover each $\bar{\mathbf{X}}_K$ from the measurement data $\mathbf{A}\bar{\mathbf{X}}_K$. For ReMBo techniques, \mathcal{P} is an i.i.d. uniform distribution in $[-1, 1]^d$.

TABLE II: Sub-Optimal Techniques

Model	Tag	Formal Description	Type
SMV	BP	Basis Pursuit, (5) with objective $\ \mathbf{x}\ _1$, see [1],[2]	Convex relaxation
	OMP	Orthogonal Matching Pursuit, see [13]	Greedy
	FOCUSS	FOcal Underdetermined System Solver, see [6]	Greedy
MMV	M-BP- ℓ_1 [14]	(6) with objective $\ \mathcal{R}_{\ell_1}(\mathbf{X})\ _1$	Convex relaxation
	M-BP- ℓ_∞ [5]	(6) with objective $\ \mathcal{R}_{\ell_\infty}(\mathbf{X})\ _1$	Convex relaxation
	M-OMP	MMV version of OMP, see [13]	Greedy
	M-FOCUSS	MMV version of FOCUSS, see [13]	Greedy
	ReMBo-BP	ReMBo with $\mathcal{S} = \text{BP}$	Convex relaxation
	ReMBo-OMP	ReMBo with $\mathcal{S} = \text{OMP}$	Greedy

4) A correct solution is announced if $\bar{\mathbf{X}}_K$ is recovered exactly up to machine precision.

The empirical recovery rate for each value of K is calculated as the percentage of correct solutions. We also collected run time data in order to qualitatively compare between the time complexity of the tested techniques. Rigorous complexity analysis and comparison are beyond the scope of this paper. Note that the selection of real-valued matrices is not mandatory and the results are also valid for complex values. However, we stick to the real-valued setting as it reproduces the setup of [13],[14]. In addition, the same empirical recovery rate is noticed when the non-zero entries of $\bar{\mathbf{X}}_K$ are drawn from a non-Gaussian distribution (e.g. uniform distribution). This behavior strengthens the conjecture that Monte-Carlo analysis is insensitive to the specific distribution of the non-zero values.

To simplify the presentation of the results, Table II 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 i th entry is equal to the ℓ_p norm of the i th row of \mathbf{X} . In the sequel we denote the `MaxIters` parameter of ReMBo based techniques in brackets, for example ReMBo-BP[1]. A default value of $\text{MaxIters} = \text{rank}(\mathbf{Y})$ is used if the brackets are omitted. This selection represents an intuitive choice, since after $\text{rank}(\mathbf{Y})$ iterations, step 5 of Algorithm 1 produces a vector \mathbf{y} that is linearly dependent in the realizations of the previous iterations. This intuition is discussed later in the results.

Note that there is a difference in deciding on a correct solution for SMV and MMV. In the latter, a solution is considered correct only when all the vectors in the matrix are recovered successfully, while in SMV a recovery of a single vector is required. Nevertheless, as both problems amount to recovering the finite support set, we plot the recovery rate curves of SMV and MMV techniques on the same scale. An alternative approach would be to adjust the SMV recovery curve so that it represents the overall success rate when the SMV technique is applied to each of the columns separately. Adjusting the results according to this approach will only intensify the improved recovery rate of ReMBo based techniques.

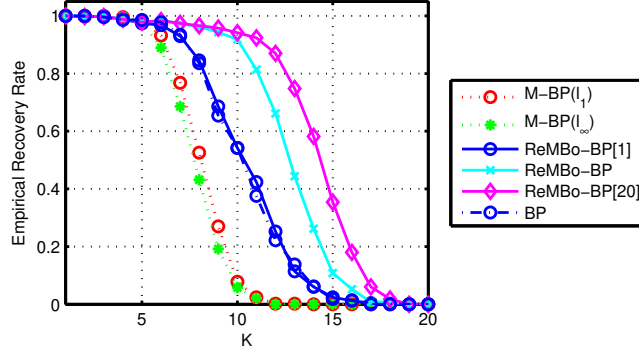


Fig. 3: Comparison of MMV techniques based on convex relaxations. The ReMBo techniques are in solid lines. As expected, the recovery curves of ReMBo-BP[1] and BP coincide.

B. Results

In Fig. 3 we compare between MMV techniques based on convex relaxation of (6). For reference we also draw the recovery rate of BP on a single measurement column. It is seen that both $M\text{-BP}(\ell_1)$ and $M\text{-BP}(\ell_\infty)$ 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. 7 the average run time of ReMBo-BP is also lower than the run time of either $M\text{-BP}(\ell_1)$ and $M\text{-BP}(\ell_\infty)$. Clearly, this simulation shows that besides the theoretical interest in the special convex relaxation of $M\text{-BP}-\ell_1$ and $M\text{-BP}-\ell_\infty$, in this example these method do not offer a practical benefit. Furthermore, the M-BP techniques require the selection of a row norm besides the standard selection of ℓ_1 norm for the final column vector. The reduction method allows to avoid this ambiguous selection by first transforming to an SMV problem.

Matching pursuit (including its variations) and FOCUSS are both greedy methods that construct the set S iteratively. These techniques are typically faster than basis pursuit based methods as seen in Fig. 7. In addition, extending the SMV version of these techniques into MMV is immediate. As opposed to convex relaxation methods, these approaches demonstrate an improved recovery rate when a joint sparsity prior is introduced. This behavior is depicted in Fig. 4. A comparison of these methods with ReMBo techniques is shown in Fig. 5. It is seen that ReMBo-OMP outperforms M-OMP and M-FOCUSS over the range $1 \leq K \leq 13$. Specifically, in the intermediate range $10 \leq K \leq 13$ it reaches a recovery rate that is approximately 10% higher than the maximal recovery rate of the non-ReMBo techniques. In addition, the run time of the ReMBo-OMP is not far from the direct greedy approaches as seen from Fig. 7.

In order to emphasize the impact of iterations, Fig. 6 depicts the recovery rate of ReMBo-BP and ReMBo-OMP for different values of MaxItrs . The recovery rate at $K = 10$ is of special interest as according to Theorem 3 $\sigma(\mathbf{A}) \geq 2K$ is required¹ to ensure that the random instances of SMV preserve the set S . For example, a single

¹According 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.

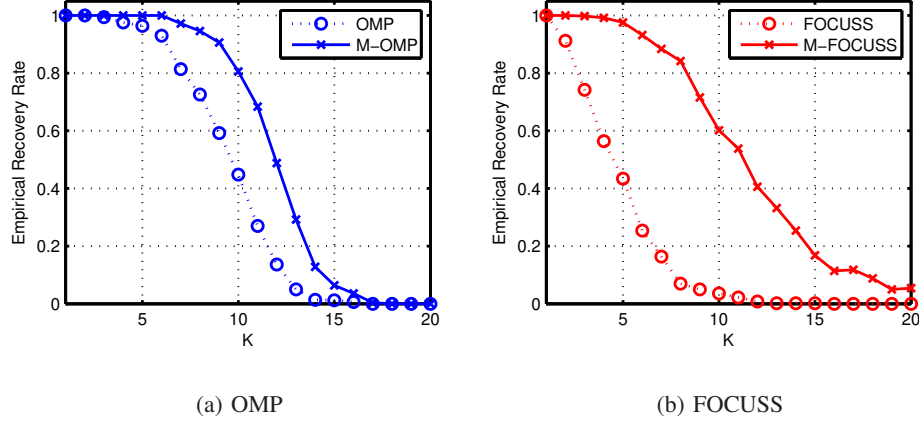


Fig. 4: The recovery rate of sequential selection techniques is demonstrated for MMV and SMV with the same number of non-zero entries per solution vector. The stopping criteria for OMP is based on the residual. The FOCUSS algorithm is designed to produce a K -sparse approximation of the solution (for this reason a ReMBo-FOCUSS method is not tested as it cannot exploit the boosting strategy).

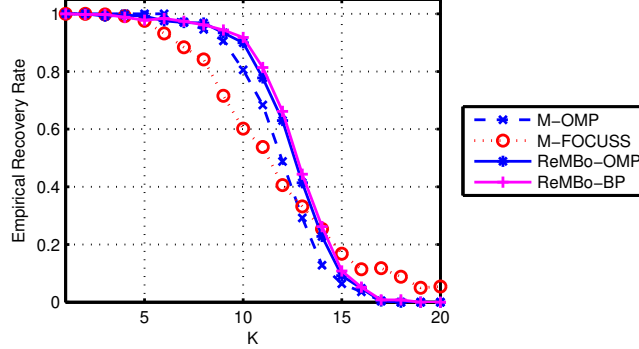


Fig. 5: A comparison between popular MMV techniques and ReMBo derived methods.

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 intuitive default selection of $\text{MaxItrs} = \text{rank}(\mathbf{Y})$. However, the condition of $K \leq 10$ is only sufficient and empirical recovery is allowed to some extent even for $K > 10$. This behavior is common to all the techniques tested here as shown in Figs. 3-6. In this range of $K > 10$, repeating the reduction process for more than $\text{rank}(\mathbf{Y})$ can be beneficial. For example, ReMBo-BP[20] yields a recovery rate of 56% for $K = 14$ instead of 25% when allowing only $\text{MaxItrs}=5$.

C. IMV Reduction vs. Discretization

We now extend the previous setup in order to simulate an IMV model by letting $d = 10000$. To discretize the IMV system, g evenly spaced columns of \mathbf{Y} are chosen resulting in an MMV system whose sparsest solution is

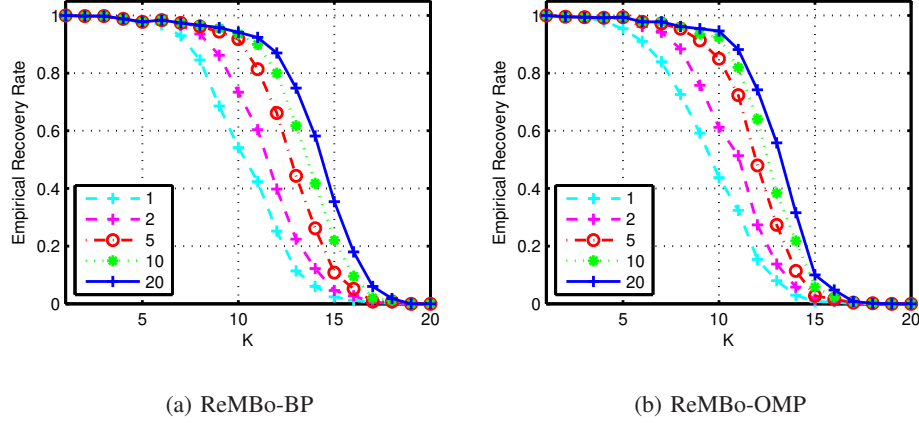


Fig. 6: The impact of boosting iterations for various selections of MaxIters.

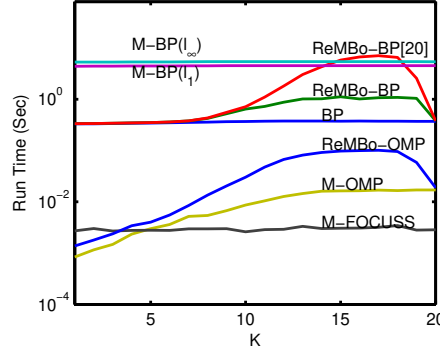


Fig. 7: Average run time of various MMV techniques.

searched, where $1 \leq g \leq 200$. Since the non-zero values are drawn randomly, interpolation of the missing columns is no useful in this setting. Instead, we consider an approximation of the non-zero location set S by taking the support of the solution matrix on the chosen grid. Finally, the entire solution set is recovered by (10)-(11). In order to capitalize on the difference between the IMV reduction flow of Fig. 1 and this discretization technique, we consider K -sparse solution matrices $\bar{\mathbf{X}}_K$ such that each non-zero row of $\bar{\mathbf{X}}_K$ has only a few non-zero entries (e.g. up to 150 non-zero values). For a fair comparison, the M-OMP technique is used for the recovery of $\bar{\mathbf{X}}_K$ in both methods.

The empirical recovery rate for several values of g is shown in Fig. 8. It is evident that a discretization technique of this type requires a grid of $g = 200$ to approach a reasonable recovery rate, which is still below the recovery rate of the IMV flow. In order to explain the superior performance of the IMV flow we plot a typical structure of a solution set in Fig. 9. It is clear that discretization may fail as it does not capture the entire information of the solution set. In contrast, our approach preserves the necessary information required for perfect reconstruction of

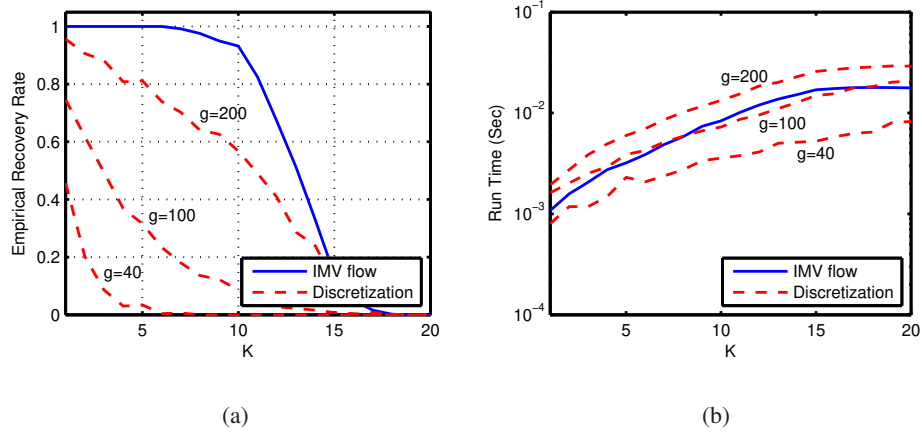


Fig. 8: A comparison of (a) the recovery rate and (b) the average run time between the IMV flow and discretization.

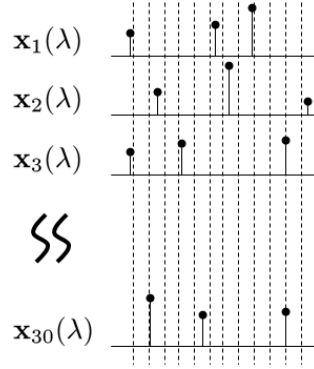


Fig. 9: A typical structure of the solution set and a grid selection. The grid cannot be synchronized with the non-zero locations. In this example, discretization technique would fail to reconstruct $\mathbf{x}_2(\lambda)$ whereas the IMV flow guarantees an exact recovery of the support set S .

the solution set, namely the non-zero location set. Furthermore, comparing the average run time of both approaches reveals that IMV is even faster than discretization having a similar recovery rate. Note that the density of the grid influences the run time of discretization methods. In the example above of $g = 200$, discretization yields an MMV system with 200 columns. The IMV flow does not have this drawback, as it follows from Lemma 1 that the matrix \mathbf{V} can be chosen such that it consists of no more than $K \leq 20$ columns.

VII. CONCLUSIONS

The essence of the reduction theorems developed in this paper is that the recovery of an arbitrary number of jointly sparse vectors amounts to solving a single sparse vector of an SMV. This result applies to the finite case of MMV and to the broader model of IMV which we introduced here. The key observation used in our developments is that the non-zero location set is the crucial information for the exact recovery of the entire solution set. We prove that this set can be recovered from a low dimensional problem rather than directly from the given high

dimensional system.

The explicit recovery problem of sparse vectors is a difficult combinatorial optimization program. Various methods to approximate the sparse solution of a given program have been previously proposed. However, to the best of our knowledge, a direct simplification of the explicit combinatorial formulation, in the way described here, was not studied so far. Furthermore, in a typical CS setting the sensing process involves randomness while the reconstruction is deterministic. The reduction method for MMV shows that randomness can also be beneficial in the reconstruction stage. In addition, popular recovery techniques have a fixed performance in terms of run time and recovery rate. In contrast, the ReMBo algorithm is tunable as it allows to trade the run time by the overall recovery rate. The simulations conducted on several ReMBo methods demonstrate this ability and affirm that these methods outperform other known techniques.

REFERENCES

- [1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [2] E. Candes, 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] D. L. Donoho and M. Elad, "Maximal sparsity representation via ℓ_1 minimization," *Proc. Natl. Acad. Sci.*, vol. 100, pp. 2197–2202, Mar. 2003.
- [4] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process. (Special Issue on Sparse Approximations in Signal and Image Processing)*, vol. 86, pp. 572–588, Apr. 2006.
- [5] —, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Process. (Special Issue on Sparse Approximations in Signal and Image Processing)*, vol. 86, pp. 589–602, Apr. 2006.
- [6] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
- [7] I. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with focuss: A recursive weighted minimum norm algorithm," *J. Electroencephalog. Clinical Neurophysiol.*, vol. 95, no. 4, pp. 231–251, Oct. 1995.
- [8] J. W. Phillips, R. M. Leahy, and J. C. Mosher, "Meg-based imaging of focal neuronal current sources," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 338–348, Jun. 1997.
- [9] B. D. Jeffs, "Sparse inverse solution methods for signal and image processing applications," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '98*, vol. 3, May 1998, pp. 1885–1888.
- [10] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [11] I. J. Fevrier, S. B. Gelfand, and M. P. Fitz, "Reduced complexity decision feedback equalization for multipath channels with large delay spreads," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 927–937, Jun. 1999.
- [12] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 374–377, Mar. 2002.
- [13] S. 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.
- [14] 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.
- [15] M. Mishali and Y. C. Eldar, "Blind multi-band signal reconstruction: Compressed sensing for analog signals," *CCIT Report no. 639, EE Dept., Technion - Israel Institute of Technology; submitted to IEEE Trans. Signal Processing*, Sep. 2007.
- [16] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, "Random filters for compressive sampling and reconstruction," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2006*, vol. 3, May 2006.

- [17] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2007*, May 2007, pp. 1959–1962.
- [18] 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.
- [19] R. Hettich and K. O. Kortanek, "Semi-infinite programming: Theory, methods, and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, Sep. 1993.
- [20] G. Still, "Generalized semi-infinite programming: Theory and methods," *European Journal of Operational Research*, vol. 119, no. 2, pp. 301–313, Dec. 1999, available at <http://ideas.repec.org/a/eee/ejores/v119y1999i2p301-313.html>.
- [21] A. F. Karr, *Probability*. New York: Springer-Verlag, 1993.
- [22] P. Billingsley, *Probability and Measure*. 3rd ed. New York: Wiley, 1995.
- [23] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," [Online]. Available: <http://www.dsp.ece.rice.edu/cs/DCS112005.pdf>.
- [24] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.
- [25] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.