Block-sparse Solutions using Kernel Block RIP and its Application to Group Lasso

Rahul Garg
IBM T.J. Watson research center
grahul@us.ibm.com

Rohit Khandekar
IBM T.J. Watson research center
rohitk@us.ibm.com

Abstract

We propose *kernel block restricted isometry property (KB-RIP)* as a generalization of the well-studied RIP and prove a variety of results. First, we present a "sum-of-norms"-minimization based formulation of the sparse recovery problem and prove that under suitable conditions on KB-RIP, it recovers the optimal sparse solution exactly. The *Group Lasso* formulation, widely used as a good heuristic, arises naturally from the Lagrangian relaxation of our formulation. We present an efficient combinatorial algorithm for provable sparse recovery under similar assumptions on KB-RIP. This result improves the previously known assumptions on RIP under which a combinatorial algorithm was known. Finally, we provide numerical evidence to illustrate that not only are our sum-of-norms-minimization formulation and combinatorial algorithm significantly faster than Lasso, they also outperforms Lasso in terms of recovery.

1 Introduction

Consider a system of linear equations of the form $y = \Phi x + e$ where $y \in \Re^m$ and $\Phi \in \Re^{m \times n}$ are given, $e \in \Re^m$ is the (unknown) error with $\|e\|_2 \leq \epsilon$, and $x \in \Re^n$ is a vector to be computed. The n components of the vector x are grouped into k blocks. A solution x is said to be s-block sparse if the components corresponding to the non-zero entries of x are contained in at most s blocks. In the special case, when all the blocks are of size unity, the problem reduces to the standard sparse regression problem that has applications in numerous domains such as compressed sensing [12], model selection [29], medical imaging [21] among others.

Although, the problem of sparse regression is NP hard in general [25], there is a rapidly growing body of literature designing efficient algorithms to solve the problem [14, 24, 4, 17] and deriving conditions under which the problem can be solved optimally [8, 6, 12]. Fundamental advancement in this field came from the formalization of the *restricted isometry property* (RIP) and results showing that if the RIP constants (δ_{2s}) of the matrix Φ satisfy some properties ($\delta_{2s} < \sqrt{2} - 1$), then the problem can be solved optimally by adding an ℓ_1 regularization penalty to the objective function.

In many cases, such as the multiple measurement vector problem [28], Group lasso [2], multi-factor ANOVA designs [31] there is a natural grouping of components of the vector x and the objective is to find a block-sparse solution. In these cases, the standard theory of compressed sensing and ℓ_1-regularization cannot be applied since the grouping typically introduces strong correlations among the columns belonging to the same group, thereby leading to poor RIP constants of the matrix Φ.

In this paper, we define *kernel block restricted isometry property* (KB-RIP) and show that under suitable conditions, optimal block-sparse solutions may be found using a sum of norms ℓ_2 penalty (also called mixed ℓ_1/ℓ_2 penalty). The Group lasso formulation [31, 2] that has been very successful in finding block-sparse solutions [30], naturally arises from the Lagrangian relaxation of our objective function. Moreover, our formulation gives a systematic method to choose the Group lasso kernels thereby providing theoretical foundations to the Group lasso based methods.

We also present an efficient combinatorial algorithm and prove that our algorithm finds the optimal block-sparse solution under certain conditions on the KB-RIP. This result provides improved conditions for

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

guaranteed recovery of sparse solutions. We also present numerical results demonstrating that, in addition to significantly better run-time performance, our algorithms provide better recovery results as compared to the other well-known algorithms for sparse recovery.

In Section 2 we present some important definitions including the definition of kernel-block RIP and relate our results to the recent results on block-based sparse recovery. In Section 3 we present our main theoretical results. Numerical results are presented in Section 4 followed by conclusions in Section 5. All the main proofs are in the supplementary material.

2 Preliminaries and Related Work

2.1 The restricted isometry property

Most of the work on sparse regression is based on the notion of the restricted isometry property.

Definition 2.1 (Restricted isometry property) *For any integer $s \geq 1$, the isometry constant δ_s of a matrix Φ is defined as the smallest real number such that the following holds for all s-sparse vectors $x \in \Re^n$:*

$$(1-\delta_s)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta_s)\|x\|_2^2. \qquad (1)$$

One of the key results in the theory of compressed sensing is that if $\delta_{2s} < \sqrt{2}-1$ [6] (or more recently $\delta_{2s} < 2(3-\sqrt{2})/7$ [16]) or $\delta_s+\delta_{2s}+\delta_{3s} < 1$ and $y = \Phi x^*$ for a s-sparse vector x^*, then x^* is also the unique optimal solution to the following ℓ_1-minimization program with $\epsilon = 0$ [9, 6].

$$\text{Minimize } \|x\|_1 \quad \text{subject to} \quad \|y-\Phi x\|_2 \leq \epsilon \qquad (2)$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ represent the ℓ_1 and ℓ_2 norm operators respectively. In "noisy" version of the problem, when x^* satisfies $\|y-\Phi x^*\|_2 \leq \epsilon$, the optimal solution to the convex program (2) satisfies $\|x-x^*\|_2 \leq C\epsilon$, where C is a small constant. In other words, the optimal solution to (2) is very close to the optimal sparse solution. The program (2) is a convex optimization problem (which reduces to a linear programming problem when $\epsilon = 0$) and can be solved in polynomial time using standard optimization techniques [19].

2.2 Failure of ℓ_1-minimization

Unfortunately, ℓ_1-minimization fails in scenarios where different parts of the unknown vector share a joint or correlated sparsity pattern. Consider, for concreteness, the *multiple measurement vector (MMV)* problem [22, 28] in which our goal is to recover a matrix X from measurements $Y = \Phi X$, for a given sampling matrix Φ. The matrix X is assumed to have at most s non-zero rows. Thus, not only each column of X is s-sparse, but in addition the non-zero columns of X share a joint sparsity pattern. The matrix equation $Y = \Phi X$ can be transformed into a linear system by stacking the columns of X and Y as $\text{vec}(Y) = (I \otimes \Phi)\text{vec}(X)$, where $\otimes$ denotes the tensor product. However note that this problem cannot directly be solved via ℓ_1-minimization since $\text{vec}(X)$ is now required to satisfy a special "block" sparsity pattern.

2.3 Group lasso

Blocking of the measurement matrix occurs naturally in many more domains. For example, in multi-factor ANOVA designs, one seeks to identify main factors and the interactions that may predict the observations. In such a case, it is desirable to group different interactions in a single block [31]. In feature selection, if some features are very similar to each other (e.g., similar proteins in a gene interaction network [18]) the corresponding columns may be grouped into a block.

For the block sparsification problems, *Group Lasso* regression models [31, 2, 23] are becoming very popular. This line of work points out the limitations of ℓ_1 penalty and proposes an ℓ_2 penalty to the standard least square objective functions when the variables are grouped into blocks. Consider a matrix $\Phi \in \Re^{m\times n}$ and a partition $\mathcal{B}$ of its n columns into k *blocks*. Let each block i contain $n_i \geq 1$ columns so that $\sum_{i=1}^k n_i = n$. Let $\Phi_i \in \Re^{m\times n_i}$ denote the matrix Φ restricted to block i. Similarly, for a vector $x \in \Re^n$, let $x_i \in \Re^{n_i}$ denote the vector x restricted to block i. Unlike ℓ_1-minimization, the group lasso models [31] are equivalent to the program:

$$\text{Minimize } \sum_{i=1}^k \|x_i\|_{K_i} \quad \text{subject to} \quad \|y-\Phi x\|_2 \leq \epsilon \quad (3)$$

or its Lagrangian relaxation

$$\min\left(\frac{1}{2}\|y-\Phi x\|_2^2 + \lambda\sum_{i=1}^k \|x_i\|_{K_i}\right)$$

where $\|x_i\|_{K_i} = \sqrt{x_i^\top K_i x_i}$ for *kernel matrices* K_i. Not knowing how to choose the kernels K_i, the practitioners often default to $K_i = I$, the identity matrix, thereby minimizing the sum of ℓ_2-norms $\sum_i \|x_i\|_2$ as part of the objective.

2.4 Block RIP

Eldar and Mishali [15] recently defined a block-restricted isometry property for finding block-sparse

solutions. Consider a matrix $\Phi \in \Re^{m \times n}$ with partition $\mathcal{B}$ of columns into k blocks. For a positive integer k, let $[k]$ denote the set $\{1, 2, \ldots, k\}$. Overloading the notation a little bit, we now let $\texttt{supp}^{\mathcal{B}}(x)$ denote the set of blocks $i \in [k]$ such that $\Phi_i x_i \neq 0$. A vector $x \in \Re^n$ is called s-block-sparse with respect to $\mathcal{B}$ if $|\texttt{supp}^{\mathcal{B}}(x)| \leq s$.

Definition 2.2 (Block RIP) *The* block isometry constant $\delta_s^{\mathcal{B}R}$ *of a matrix* Φ *with respect to block partitioning* $\mathcal{B}$ *is the smallest number such that*

$$(1 - \delta_s^{\mathcal{B}R})\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s^{\mathcal{B}R})\|x\|_2^2$$

holds for all s-block-sparse $x \in \Re^n$.

Extending the proof of Candes [6], they showed that if the block isometry constant of Φ satisfies $\delta_{2s}^{\mathcal{B}R} < \sqrt{2} - 1$, the optimum solution to the Group lasso program (3) with $K_i = I$ for all i is also the optimum s-block-sparse optimum solution when $\epsilon = 0$. In case $\epsilon > 0$, the optimal solution to (3) is very close to the optimal sparse solution.

In a related work, Baraniuk et al. [3] defined a model-based restricted isometry property which is a generalization of the block-RIP. For a specific case when model-based RIP becomes equivalent to the block-RIP, they showed that their combinatorial algorithm called block-based CoSaMP finds the optimal s-block-sparse solution when $\delta_{4s}^{\mathcal{B}R} \leq 0.1$.

Our following result shows that the block-RIP is very sensitive to the conditioning of the individual blocks Φ_i of the matrix Φ. So, if the individual blocks of the matrix Φ are poorly conditioned (which is expected if the columns of a blocks are similar), the block-RIP based algorithms become unsuitable for recovery.

For a symmetric matrix $A \in \Re^{n \times n}$, let $\lambda_{\max}(A)$ denote the maximum absolute eigenvalue of A.

Lemma 2.1 *Consider a matrix* $\Phi \in \Re^{m \times n}$ *with partition* $\mathcal{B}$ *of columns into* k *blocks* $\Phi_i \in \Re^{m \times n_i}$ *for* $i \in [k]$. *Let*

$$\lambda = \max_{i=1}^{k} \{\lambda_{\max}(\Phi_i^\top \Phi_i)\}$$

be the maximum absolute value of any eigenvalue of $\Phi_i^\top \Phi_i$ *for any block* Φ_i. *Then for any* $s \geq 1$, *we have* $\delta_s^{\mathcal{B}R} \geq \lambda - 1$.

Proof. Since $\delta_{s+1}^{\mathcal{B}R} \geq \delta_s^{\mathcal{B}R}$ for all $s \geq 1$, it is enough to show that $\delta_1^{\mathcal{B}R} \geq \lambda - 1$. Suppose the maximum in the definition of λ is achieved for $i = 1$. Thus $\lambda = \lambda_{\max}(\Phi_1^\top \Phi_1)$. Let $v \in \Re^{n_1}$ be the unit-norm eigenvector corresponding to the maximum eigenvalue. Define a vector $x \in \Re^n$ as $x_1 = v$ and $x_i = 0$ for all $i \neq 1$. Note that $|\texttt{supp}^{\mathcal{B}}(x)| = 1$, i.e., x is 1-block-sparse. Now it is easy to see that

$$\frac{\|\Phi x\|_2^2}{\|x\|_2^2} = \frac{\|\Phi_1 x_1\|_2^2}{\|x_1\|_2^2} = \frac{v^\top \Phi_1^\top \Phi_1 v}{v^\top v} = \lambda.$$

The lemma therefore follows from the definition of block-RIP for this case. A similar argument holds when the maximum in the definition of λ is achieved for other blocks. ∎

Any recovery result using block RIP requires the maximum absolute eigenvalue (and analogously the minimum eigenvalue) to be close to one.

2.5 Kernel block restricted isometry property

We define the kernel block restricted isometry property which is invariant to any linear transformation of the blocks and therefore is insensitive to the conditioning of the individual blocks.

Definition 2.3 (Kernel Block RIP) *The* kernel block isometry constant $\delta_s^{\mathcal{B}}$ *of a matrix* Φ *with respect to block partitioning* $\mathcal{B}$ *is the smallest number such that*

$$(1 - \delta_s^{\mathcal{B}}) \sum_{i=1}^{k} \|\Phi_i x_i\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s^{\mathcal{B}}) \sum_{i=1}^{k} \|\Phi_i x_i\|_2^2$$

holds for all s-block-sparse $x \in \Re^n$.

When all blocks are singletons and all columns Φ_i have unit ℓ_2 norm, the above definition reduces to that of the well-known isometry constants.

We argue that kernel block RIP captures the inter-block interactions better than the block RIP. It is easy to see that if the columns in Φ_i are orthogonal and have unit ℓ_2 norm, we have $\|\Phi_i x_i\|_2 = \|x_i\|_2$. In such a case, $\sum_{i=1}^{k} \|\Phi_i x_i\|_2^2 = \sum_{i=1}^{k} \|x_i\|_2^2 = \|x\|_2^2$ and the kernel block RIP becomes equivalent to the block RIP. However if the columns of Φ_i are not orthonormal, the ratio $\|x_i\|_2 / \|\Phi_i x_i\|_2$ can be as large as the condition number of Φ_i, i.e., it depends on how orthogonal the columns of Φ_i are. Even worse, this ratio can be infinite if the columns of Φ_i are linearly dependent, e.g., if Φ_i has a repeated column. In other words, the block isometry constants depend on the factors arising due to ill conditioning of Φ_i. We remove this dependence in kernel block RIP. Since the terms $\|\Phi_i x_i\|_2$ appear on the leftmost and rightmost sides in definition 2.3, the kernel block isometry constants depend only on the inter-block interactions, abstracting away the effects of ill conditioning of individual blocks. This is formalized by the following result.

Lemma 2.2 *Consider a matrix $\Phi \in \Re^{m\times n}$ with partition $\mathcal{B}$ of columns into k blocks $\Phi_i \in \Re^{m\times n_i}$ for $i \in [k]$. Let $T_i \in \Re^{n_i\times n_i}$ be any invertible matrix for $i \in [k]$. Define a matrix $\Psi \in \Re^{m\times n}$ with partition $\mathcal{B}$ of columns into k blocks $\Psi_i = \Phi_i T_i \in \Re^{m\times n_i}$ for $i \in [k]$. Then for any $s \geq 1$, the Kernel block RIP constant $\delta_s^{\mathcal{B}}(\Phi)$ of Φ is exactly same as the Kernel block RIP constant $\delta_s^{\mathcal{B}}(\Psi)$ of Ψ.*

Proof. Since T_i is invertible, for any $u \in \Re^{n_i}$, there exists $v \in \Re^{n_i}$ such that $u = T_i v$. Thus for any s-block-sparse vector x, there exists an s-block sparse vector y such that $\Phi_i x_i = \Psi_i y_i$ for $i \in [k]$, and vice versa. The lemma now follows from the definition of the Kernel block RIP constants. ∎

Unlike block RIP, the kernel block RIP constants of a matrix Φ may be small even if some of its blocks are not full rank. In this case, if the support of a solution x contains at least one such block, then there are infinite block sparse solutions. All these solutions may be obtained by adding vectors in the null space of the non-zero blocks, to a block sparse solution. To find one such solution, the dependent columns of each of the sub-matrices Φ_i may be removed until all of them become full rank. This makes the solution unique which may be found by our algorithms described in the later sections. Once a block sparse solution is found, all the block sparse solutions may be obtained by adding vectors in the null space of the non-zero blocks. Therefore, in rest of this paper, we assume that each of the submatrix Φ_i is full rank.

Consider the MMV problem of computing a matrix X with at most s non-zero rows such that $Y = \Phi X$. It is easy to see that with respect to the appropriate blocking $\mathcal{B}$ of the columns, the block isometry constants of the matrix $I \otimes \Phi$ are identical to the isometry constants of Φ. Therefore a variety of known results regarding RIP [10, 7] directly apply for block RIP of the underlying matrix in MMV.

In compressed sensing, the kernel block RIP gives more flexibility in designing the encoding matrix Φ. In application areas such as MRI [21, 20] a truly random sampling of phase space, which is almost guaranteed to generate matrix Φ satisfying RIP [10, 8], is very demanding on the magnetic field gradient coils of the system [20]. It is desirable to design the phase space sampling scheme that does not require very fast changes to the magnetic field gradients. The kernel block RIP requires fewer constraints in designing the MRI acquisition sequence.

3 Our Results

3.1 Exact convex relaxation

We now state our first main result giving an equivalence between the ℓ_0 and the following ℓ_2 minimization problem.

Theorem 3.1 (Exact convex relaxation) *Let Φ be a matrix satisfying $\delta_{2s}^{\mathcal{B}} < \sqrt{2} - 1$ for an integer s and blocks $\mathcal{B}$ and assume that there exists an s-block-sparse x such that $y = \Phi x$. Then the problem of computing such a solution is equivalent to solving the following ℓ_2-minimization problem:*

$$\min \sum_{i=1}^{k} \|\Phi_i x_i\|_2 \quad \textit{subject to} \quad y = \Phi x. \tag{4}$$

Note a crucial difference in this theorem – the previous results (except [15]) have shown equivalence between ℓ_0 and ℓ_1 minimizations under RIP. For kernel block RIP, however, it turns out that the "natural" ℓ_1-minimization $\min\{\sum_{i=1}^{k} \|\Phi_i x_i\|_1 \mid y = \Phi x\}$ is *not* equivalent to the ℓ_0-minimization. However when all the blocks are singletons and have unit norm, the kernel block RIP becomes identical to RIP and the ℓ_0, ℓ_1, and ℓ_2 minimizations all become equivalent.[1] The proof of Theorem 3.1, which is based on a reduction to the block RIP and a theorem of Eldar and Mishali [15], is outlined in the supplementary material.

3.2 Relationship with group lasso

Note that the group lasso program (3) with $\epsilon = 0$ becomes equivalent to our program (4) if we define kernels as $K_i = \Phi_i^\top \Phi_i$. Thus our work gives a structured method and a theoretical justification to select the kernels K_i used in group lasso based regression. Moreover, the combinatorial Algorithm 1 proposed below may also be a useful, computationally efficient alternative to the convex optimization problem of group lasso for model selection with grouped variables.

3.3 A fast combinatorial algorithm

The ℓ_2-minimization problem (4) is a convex program which may be solved in polynomial time using standard convex optimization techniques [19]. Despite polynomial time guarantee, the run-time of these solvers may be unacceptably large especially for large

[1] Note that when all blocks are singletons, the ℓ_2-minimization (4) is *not* equivalent to the "usual" ℓ_2-minimization $\min\{\|x\|_2 \mid y = \Phi x\}$ but is equivalent to $\min\{\|x\|_1 \mid y = \Phi x\}$ assuming that all columns of Φ have unit norm.

problem sizes. Our next contribution is a combinatorial algorithm to solve the ℓ_0 minimization to an arbitrary precision. Let $\phi = (1+\sqrt{5})/2 \approx 1.618$ denote the *golden ratio*.

Our combinatorial algorithm is presented in Algorithm 1. This is very similar to the iterative hard thresholding algorithm [4, 17] which updates the current solution by moving along the gradient of the least-square error objective function. The hard thresholding step of [4] is replaced by the block-based hard thresholding function defined as follows.

Definition 3.1 *Let $H_s^{\mathcal{B}} : \Re^n \to \Re^n$ be a function that sets all but s blocks i with the largest values of $\|\Phi_i x_i\|$ to zero. More precisely, for $x \in \Re^n$, let π be a permutation of $[k]$ such that $\|\Phi_{\pi(1)} x_{\pi(1)}\| \geq \cdots \geq \|\Phi_{\pi(k)} x_{\pi(k)}\|$. Then the vector $H_s^{\mathcal{B}}(x)$ is a vector x' where $x'_{\pi(i)} = x_{\pi(i)}$ for $i \leq s$ and $x'_{\pi(i)} = 0$ for $i \geq s+1$.*

Algorithm 1 *Block Iterative Hard Thresholding (Block IHT)* Algorithm

Initialize $x \leftarrow 0$.
for t iterations **do**
  $\hat{x} \leftarrow (\hat{x}_1, \ldots, \hat{x}_k)$ where,

$$\hat{x}_i = x_i + \left(\Phi_i^\top \Phi_i\right)^{-1} \Phi_i^\top (y - \Phi x) \qquad \text{for all } i \in [k].$$

  $x \leftarrow H_s^{\mathcal{B}}(\hat{x})$.
end for
Output x.

Let $\lambda_{\min}(A)$ denote the minimum absolute eigenvalue of the matrix A. Let $\lambda_{\min} = \min_{i=1}^{k} \{\lambda_{\min}(\Phi_i^\top \Phi_i)\}$.

Theorem 3.2 *Let Φ be a matrix with blocks $\mathcal{B}$ satisfying*

$$\delta_{2s}^{\mathcal{B}} < \frac{1}{\sqrt{3}\cdot\phi} \approx 0.357 \qquad \text{or} \qquad \delta_{3s}^{\mathcal{B}} < \frac{1}{\phi} \approx 0.618,$$

and $y = \Phi x^*$ for some s-block-sparse vector $x^* \in \Re^n$. In t iterations, Algorithm 1 computes an s-block-sparse vector $x \in \Re^n$ satisfying

$$\|x^* - x\|_2^2 \leq \frac{2\|y\|_2^2}{\lambda_{\min}} \cdot \alpha^t \qquad \text{and} \qquad \|y - \Phi x\|_2^2 \leq 2\|y\|_2^2 \cdot \alpha^t$$

where

$$\alpha = \left[\phi \cdot \min\{\sqrt{3}\cdot\delta_{2s}^{\mathcal{B}}, \delta_{3s}^{\mathcal{B}}\}\right].$$

The algorithm initially computes and stores the inverses $(\Phi_i^\top \Phi_i)^{-1}$ for $i \in [k]$, and computes one matrix-vector product with the matrices Φ, $\Phi_i^\top$, and $(\Phi_i^\top \Phi_i)^{-1}$ in each iteration.

The proof of this theorem is given in the supplementary material.

Thus, in $\left\lceil C \cdot \log\left(\frac{2\|y\|_2^2}{\epsilon \cdot \lambda_{\min}}\right)\right\rceil$ iterations the algorighm finds a solution satisfying $\|x^* - x\|_2^2 \leq \epsilon$, where $C = -1/\log(\alpha)$. Note that logrithmic dependence between the number of iterations, the desired accucary ϵ and the minimum absolute condition number $\lambda_{\min}$ implies that a small number of iterations is needed to get very close to the optimal solution. The above combinatorial approach not only improves the running time significantly, but also improves the conditions on the kernel block RIP constants $\delta_{2s}^{\mathcal{B}}$ and $\delta_{3s}^{\mathcal{B}}$ for which problem can be solved combinatorially, *even when all the blocks are singletons*, i.e., even in the special case of RIP constants.

4 Numerical Results

In this section we compare the performance of four algorithms on a class of randomly generated matrices. These algorithms include

- Block IHT: Algorithm 1 proposed in this paper.
- Kernel block L2: The group lasso with kernels proposed in this paper (Section 3.2),
- Block L2: The group lasso with identity kernel matrices (see [15]),
- Lasso/LARS: The classical lasso regression using LARS [14],

Let x^* represent the optimal solution and x represent the solution found by the algorithm. Recall that $\texttt{supp}(x)$ represents the support, i.e., the set of indices of non-zero components, of the vector x. The following performance metrics are used for the comparison.

- Precision: The fraction of non-zero entries in x that are also non-zero in x^*, i.e., the precision of x is $|\texttt{supp}(x) \cap \texttt{supp}(x^*)|/|\texttt{supp}(x)|$.
- Recall.: The fraction of non-zero entries in x^* that are also non-zero in x, i.e., the recall of x is $|\texttt{supp}(x) \cap \texttt{supp}(x^*)|/|\texttt{supp}(x^*)|$.
- Normalized error: The ℓ_2 norm of the error vector divided by the ℓ_2 norm of x^*, i.e., the normalized error of x is $\|x - x^*\|_2/\|x^*\|_2$.
- Normalized residue: The ℓ_2 norm of the error in the observation vector divided by the ℓ_2 norm of y, i.e., the normalized residue of x is $\|y - \Phi x\|_2/\|y\|_2$.
- CPU time taken: The time taken to compute the solution.

Each random instance is generated as follows. Let $N(0,1)$ denote the normal distribution with zero mean and unit variance. Let $U(d)$ denote the distribution on vectors in $\Re^d$ where each entry is independently drawn from $N(0,1)$. We obtain an $m \times n$ matrix Φ as follows. First, the columns of Φ are grouped into equal-sized blocks of size b. Half of the blocks are categorized as "uncorrelated blocks" and the other half as "correlated blocks". The columns in the uncorrelated blocks are independently drawn from $U(m)$. For the correlated blocks, the first column is independently drawn from $U(m)$. Each subsequent column in that block is generated by adding the first column to a constant α times a vector independently drawn from $U(m)$. All the columns are subsequently scaled to have unit ℓ_2 norm.

To generate an s-block-sparse vector x^*, a set of s blocks is selected randomly. For each of the selected blocks Φ_i, a random unit norm vector, uniformly distributed over the column space of Φ_i is picked. This vector is scaled by a scalar drawn from $N(0,1)$ to obtain y_i. The corresponding 1-block-sparse x_i such that $y_i = \Phi_i x_i$, was computed. The vectors y and x^* are given by $y = \sum_i y_i$ and $x^* = \sum_i x_i$.

All the algorithms were implemented using the Matlab software. The implementation of "Lasso/LARS" was taken from the SparseLab package [13]. The implementation of "Block L2" regression (or group lasso) was taken from [1]. We modified it to implement "Kernel block L2". We also implemented "Block IHT" using Matlab.

For the results presented here, we work with $m = 500, n = 1000, b = 2, \alpha = 0.1$. The sparsity parameter s takes seven different values ranging from 2 to 100. For each of these values, one hundred random instances of the problem are generated. Each instance is solved using each of the four methods described earlier and the performance metrics for each run are computed. To ensure a reasonable run-time, the maximum number of iterations in each of the algorithms is set to $100s$. Even with this setting, Lasso/LARS takes a large amount of time for large values of s. So, the maximum number of iterations for Lasso/LARS is set to $5s + 100$.

Figure 1 compares the precision of all the algorithms for different values of s. Overall, the precision of Block IHT is significantly better than the other methods. When $s \leq 10$, Block IHT always recovers the correct solution for all the hundred random instances of the problem. In contrast, the mean precision of all the other algorithms is less than 0.4 even for small values of s. Figure 2 compares the recall performance of these algorithms. The recall of Block IHT is consistent

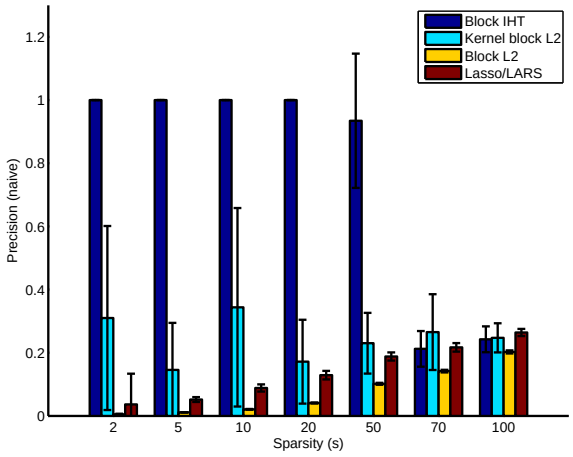

Figure 1: Precision without post-processing

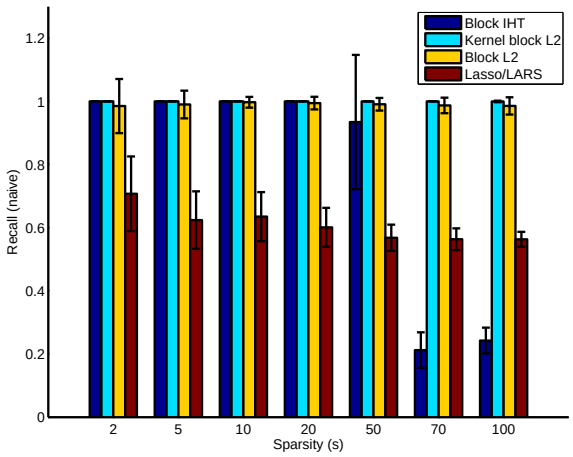

Figure 2: Recall without post-processing

with its precision. However, the mean recall values of all the other algorithms are significantly higher than their corresponding precision values. The mean recall is almost always 1 for Kernel block L2 and is very close to one for Block L2 algorithm.

It turns out that all the algorithms except Block IHT return a large number of non-zero entries in x. A good fraction of these entries are very close to zero. For a fair comparison of the algorithms we modify the final solution by choosing top s blocks according to the largest ℓ_2 norms. The problem is then solved exactly (by the method of least squares) for the matrix Φ restricted to these s blocks. This solution has exactly s non-zero blocks and hence precision and recall values are identical. All the performance metrics are recomputed for the new solution.

Figures 3 and 4 respectively compare the precision and normalized error after post-processing. Now, Kernel block L2 always outperforms all the other algorithms

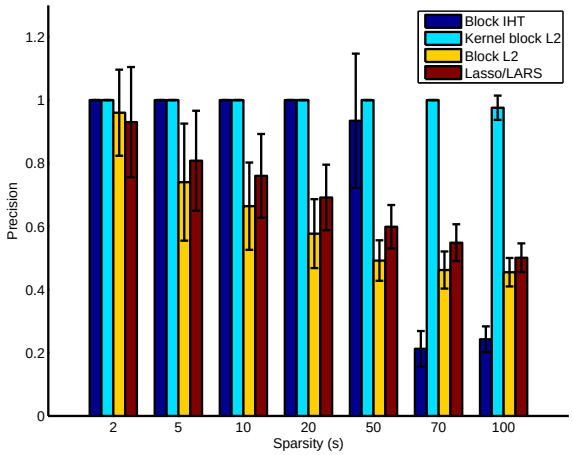

Figure 3: Precision (= recall) after post-processing

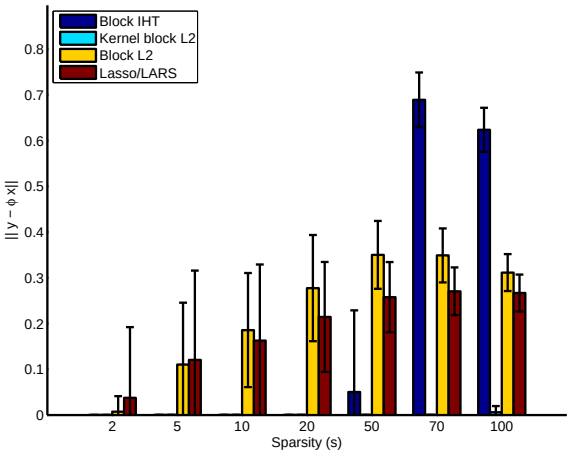

Figure 5: Normalized residue ($\|y - \Phi x\|/\|y\|$)

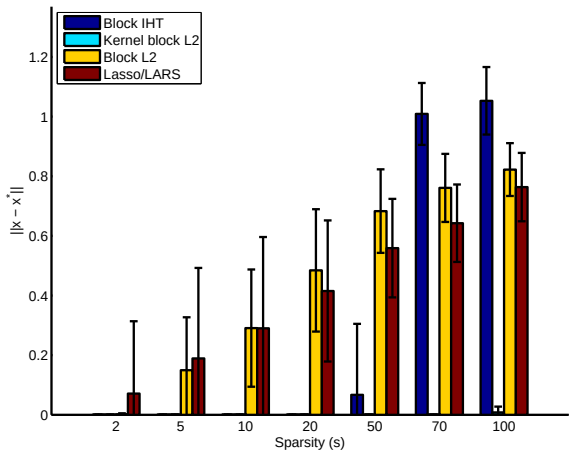

Figure 4: Normalized error after post-procesing

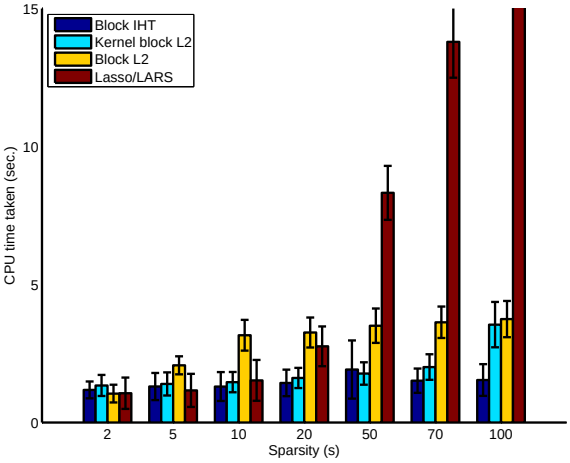

Figure 6: (f) CPU time taken (sec.)

on both the metrics, and significantly so for large values of s. The Block IHT performs almost perfect recovery as long as the sparsity parameter s is at most 50 whereas the recovery properties of Block L2 and Lasso/LARS are worse. Figure 5 shows the normalized residues for all the algorithms. It is instructive to note that the normalized residues are always less than the corresponding normalized errors. In fact, for Block L2 and Lasso, there were many instances when the residue was close to zero, but the normalized error was significantly high. In these instances, the algorithms Block L2 and Lasso appear to have found the correct solution, whereas in reality the solution was far from the correct one. Once again, Kernel block L2 outperforms all the other algorithms significantly.

Finally Figure 6 compares the run-time of the algorithms. When the sparsity parameter is small, the run-times of all the algorithms are comparable. However, as s is increased, the run-times of all the algorithms except Block IHT increase. In fact, the run-time of Lasso becomes many orders of magnitude higher than the rest for $s = 100$. The fastest algorithm is Block IHT followed by Kernel block L2 and Block L2, respectively.

For all the instances the recovery properties of our algorithm, Kernel block L2, are found to be better than the rest of the algorithms. Its run-time is better than the state-of-the-art algorithms Lasso/LARS and Block L2, next only to the proposed combinatorial algorithm Block IHT. For small values of s, the recovery properties of Block IHT are very good too.

5 Conclusions

In this paper we presented two algorithms, Kernel block L2 and Block IHT, to solve the sparse regression problem that are most suitable for instances where columns of the measurement matrix Φ can be naturally

grouped into blocks. Such instances arise naturally in many settings such as MMV or regression problems where groups of feature vectors are correlated.

Both our algorithms significantly outperform the state-of-the art algorithms (lasso and group-lasso) in terms of recovery properties as well as run-times. Our first algorithm, Kernel block L2 (Theorem 3.2), recovers correct solution for a larger class of problems as compared to lasso, group-lasso, and Block IHT. On the other hand, the run-time of Block IHT (Algorithm 1) is significantly better than that of lasso, group-lasso, and Kernel block L2 for large problems. The proposed algorithm Kernel block L2 may be viewed as an instance of the group-lasso algorithm with suitably chosen kernel matrices. Most of current literature on group-lasso is silent about the choice of kernel matrices and resort to simply block ℓ_2 norms. Our work gives a way to choose the kernel matrices in the group-lasso setting. Moreover, we give theoretical conditions based on the property of "kernel block-RIP", under which our algorithms are guaranteed to find the correct solution. Matrices satisfying kernel block RIP arise naturally in problems such as MMV. These conditions cover a much larger set of problems than the existing consistency conditions known for group lasso [30, 26, 5, 2, 27] and lasso [14, 11].

References

[1] F. Bach. Group Lasso package. http://www.di.ens.fr/ fbach/grouplasso/index.htm, 2008.

[2] F. R. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, C. Hegde, and M. B. Wakin. Model-based compressive sensing. *IEEE Transactions in Information Theory*, 2010.

[4] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. Preprint, 2008.

[5] H. Bondell and B. Reich. Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR. *Biometrics*, 2008.

[6] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences, Paris*, 1(346):589–592, 2008.

[7] E. J. Candès and J. Romberg. Practical signal recovery from random projections. In *SPIN Conference on Wavelet Applications in Signal and Image Processing*, 2004.

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

[9] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[10] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[12] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, 2006.

[13] D. Donoho and Others. SparseLab: Seeking sparse solutions to linear systems of equations. http://sparselab.stanford.edu/, 2009.

[14] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(1):407–499, 2004.

[15] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inf. Theor.*, 55(11):5302–5316, 2009.

[16] S. Foucart and M.-J. Lai. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26(3):395–407, 2009.

[17] R. Garg and R. Khandekar. Gradient descent with sparsication: An iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of International Conference on Machine Learning (ICML)*, 2009.

[18] Gene interaction network. http://gin.ncibi.org/, 2008.

[19] M. Grotschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[20] M. Lustig. Sparse MRI. Ph.D Thesis, Stanford University, 2008.

[21] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2005.

[22] M. Mishali and Y. C. Eldar. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. *IEEE Trans. Signal Process.*, 56(10):4692–4702, 2008.

[23] Y. Nardi and A. Rinaldo. On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2:605–633, 2008.

[24] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2008.

[25] T. Neylon. *Sparse solutions for linear prediction problems*. PhD thesis, Courant Institute, New York University, 2006.

[26] M. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 2008.

[27] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, Series B*, 2009.

[28] J. A. Tropp. Algorithms for simultaneous sparse approximation: Parts I and II. *Signal Process.*, 86:572–602, 2006.

[29] M. J. Wainwright, P. Ravikumar, and J. D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 1465–1472. MIT Press, Cambridge, MA, 2006.

[30] L. Wang, G. Chen, and H. Li. Group SCAD regression analysis for microarray time course gene expression data. *Bioinformatics*, 2007.

[31] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.