C-HiLasso: A Collaborative Hierarchical Sparse Modeling Framework

Pablo Sprechmann,[1†] Ignacio Ramirez,[1†] Guillermo Sapiro[1] and Yonina C. Eldar[2]
[1]University of Minnesota and [2]Technion

Abstract

Sparse modeling is a powerful framework for data analysis and processing. Traditionally, encoding in this framework is performed by solving an ℓ_1-regularized linear regression problem, commonly referred to as *Lasso* or *basis pursuit*. In this work we combine the sparsity-inducing property of the Lasso model at the individual feature level, with the block-sparsity property of the *Group Lasso* model, where sparse groups of features are jointly encoded, obtaining a sparsity pattern hierarchically structured. This results in the *Hierarchical Lasso (HiLasso)*, which shows important practical modeling advantages. We then extend this approach to the collaborative case, where a set of simultaneously coded signals share the same sparsity pattern at the higher (group) level, but not necessarily at the lower (inside the group) level, obtaining the collaborative HiLasso model *(C-HiLasso)*. Such signals then share the same active groups, or classes, but not necessarily the same active set. This model is very well suited for applications such as source identification and separation. An efficient optimization procedure, which guarantees convergence to the global optimum, is developed for these new models. The underlying presentation of the new framework and optimization approach is complemented with experimental examples and theoretical results regarding recovery guarantees for the proposed models.

I. Introduction and Motivation

Sparse signal modeling has been shown to lead to numerous state-of-the-art results in signal processing, in addition to being very attractive at the theoretical level. The standard model assumes that a signal can be efficiently represented by a sparse linear combination of atoms from a given or learned dictionary. The selected atoms form what is usually referred to as the *active set*, whose cardinality is significantly smaller than the size of the dictionary and the dimension of the signal.

In recent years, it has been shown that adding structural constraints to this active set has value both at the level of representation robustness and at the level of signal interpretation (in particular when the active set indicates some physical properties of the signal); see [1], [2], [3] and references therein. This leads to *group* or *structured* sparse coding, where instead of considering the atoms as singletons, the atoms are grouped, and a few groups are active at a time. An alternative way to add structure (and robustness) to the problem is to consider the simultaneous encoding of multiple signals, requesting that they all share the same active set. This is a natural collaborative filtering approach to sparse coding; see, for example, [4], [5], [6], [7], [8], [9].

In this work we extend these models in a number of directions. First, we present a hierarchical sparse model, where not only a few (sparse) groups of atoms are active at a time, but also each group enjoys internal sparsity.[1] At the conceptual level, this means that the signal is represented by a few groups

[†] P. S. and I. R. contributed equally to this work.

[1] While we here consider only 2 levels of sparsity, the proposed framework is easily extended to multiple levels.

(classes), and inside each group only a few members are active at a time. A simple example of this is a piece of music (numerous applications in genomics and image processing exist as well), where only a few instruments are active at a time (each instrument is a group), and the sound produced by each instrument at each instant is efficiently represented by a few atoms of the sub-dictionary/group corresponding to it. Thereby, this proposed hierarchical sparse coding framework permits to efficiently perform source identification and separation, where the individual sources (classes/groups) that generated the signal are identified at the same time as their representation is reconstructed (via the sparse code inside the group). An efficient optimization procedure, guaranteed to converge to the global optimum, is proposed to solve the hierarchical sparse coding problems that arise in our framework. Theoretical recovery bounds are derived for this hierarchical sparse model (*HiLasso*).

Then, we go one step beyond this. Continuing with the above example, if we know that the same few instruments will be playing simultaneously during different passages of the piece, then we can assume that the active groups at each instant, within the same passage, will be the same. We can then exploit this information by applying the new hierarchical sparse coding approach in a collaborative way, enforcing that the same groups will be active at all instants within a passage (since they are of the same instruments and then efficiently representable by the same sub-dictionaries), while allowing each group for each music instant to have its own unique internal sparsity pattern (depending on how the sound of each instrument is represented at each instant). We propose a collaborative hierarchical sparse coding framework addressing exactly this *(C-HiLasso)*.[2] An efficient optimization procedure for this case is derived as well. For this case, we comment on results regarding the correct recovery of the underlying active groups.

Our proposed optimization technique for both sparse coding problems combines the *Sparse Reconstruction by Separable Approximation* (SpaRSA) [10], with the *Alternating Direction Method of Multipliers* (ADMOM) [11], a general purpose optimization tool that has been successfully employed to efficiently solve ℓ_1-constrained regularization problems [12], [13], [14]. Our algorithm iteratively alternates between a scalar thresholding at the coefficient level and a vector thresholding at the group level, naturally yielding to the desired hierarchical sparsity patterns in the solutions and converges to the global optimum.

The rest of the paper is organized as follows: Section II provides an introduction to traditional sparse modeling and presents our proposed HiLasso and C-HiLasso models. After introducing the models, we discuss their relationship with the recent works of [2], [15], [16], [17], [18], [19] is detailed. In Section III we describe the optimization techniques applied to solve the resulting sparse coding problems. Recovery guarantees for HiLasso in the noiseless setting are developed in Section IV. We also comment on existing results regarding correct recovery of group-sparse patterns in the collaborative case. Experimental results and simulations are given in Section V, and finally concluding remarks are presented in Section VI.

II. Collaborative Hierarchical Coding

A. Background: Lasso and Group Lasso

Assume we have a set of data samples $\mathbf{x}_j \in \mathbb{R}^m, j = 1, \ldots, n$, and a dictionary of p atoms in $\mathbb{R}^m$, assembled as a matrix $\mathbf{D} \in \mathbb{R}^{m \times p}$, $\mathbf{D} = [\mathbf{d}_1 \mathbf{d}_2 \ldots \mathbf{d}_p]$. Each sample $\mathbf{x}_j$ can be written as $\mathbf{x}_j = \mathbf{D}\mathbf{a}_j + \epsilon$, $\mathbf{a}_j \in \mathbb{R}^p$, $\epsilon \in \mathbb{R}^m$, that is, as a linear combination of the atoms in the dictionary $\mathbf{D}$ plus some perturbation ϵ, satisfying $\|\epsilon\|_2 \ll \|\mathbf{x}_j\|_2$. The basic underlying assumption in sparse modeling is that, for all or most j, the "optimal" reconstruction $\mathbf{a}_j$ has only a few nonzero elements. Formally, if we define

[2] Note that different recordings can also have different instruments, so some of them will share the same groups while not necessarily all of them will be exactly the same.

the ℓ_0 cost as the pseudo-norm counting the number of nonzero elements of $\mathbf{a}_j$, $\|\mathbf{a}_j\|_0 = |\{k : a_{kj} \neq 0\}|$, then we expect that $\|\mathbf{a}_j\|_0 \ll p$ and $\|\mathbf{a}_j\|_0 \ll m$ for all or most j.

The ℓ_0 optimization is non-convex and known to be NP-hard. To determine $\mathbf{a}_j$ in practice, a multitude of efficient algorithms have been proposed, which achieve high recovery rates. The ℓ_1-minimization method is the most extensively studied recovery technique. In this approach, the non-convex ℓ_0 norm is replaced by the convex ℓ_1 norm, leading to

$$\min_{\mathbf{a}\in\mathbb{R}^p} \|\mathbf{a}\|_1 \quad \text{s.t.} \quad \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 \leq \epsilon. \tag{II.1}$$

The use of general purpose or specialized convex optimization techniques allows for efficient reconstruction using this strategy. The above approximation is known as the Lasso [20] or basis pursuit [21], [22]. A popular variant is to use the unconstrained version

$$\min_{\mathbf{a}\in\mathbb{R}^p} \frac{1}{2} \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 , \tag{II.2}$$

where λ is an appropriate parameter value, usually found by cross-validation.

The fact that the $\|\cdot\|_1$ regularizer induces sparsity in the solution $\mathbf{a}_j$ is desirable not only from a regularization point of view, but also from a model selection perspective, where one wants to identify the relevant features or factors (atoms) that conform each sample $\mathbf{x}_j$. In many situations, however, the goal is to represent the relevant factors not as singletons but as groups of atoms. For a dictionary of p atoms, we define groups of atoms through their indexes, $G \subseteq \{1, \ldots, p\}$. Given a group G of atoms from a dictionary $\mathbf{D}$, we denote the subdictionary formed by them as $\mathbf{D}_G$, and the corresponding set of linear reconstruction coefficients as $\mathbf{a}_G$. Define $\mathcal{G} = \{G_1, \ldots, G_{|\mathcal{G}|}\}$ to be a partition of $\{1, \ldots, p\}$.[3] In order to perform model selection at the group level (relative to the partition $\mathcal{G}$), the Group Lasso problem was introduced in [1],

$$\min_{\mathbf{a}\in\mathbb{R}^p} \frac{1}{2} \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 + \lambda\psi_{\mathcal{G}}(\mathbf{a}), \tag{II.3}$$

where $\psi_{\mathcal{G}}$ is the Group Lasso regularizer defined in terms of $\mathcal{G}$ as $\psi_{\mathcal{G}}(\mathbf{a}) = \sum_{G\in\mathcal{G}} \|\mathbf{a}_G\|_2$. The function $\psi_{\mathcal{G}}$ can be seen as an ℓ_1 norm on Euclidean norms of the vectors formed by coefficients belonging to the same group $\mathbf{a}_G$. This is a generalization of the ℓ_1 regularizer, as the latter arises from the special case $\mathcal{G} = \{1, 2, \ldots, p\}$ (the groups are singletons), and as such, its effect on the groups of $\mathbf{a}$ is also a natural generalization of the one obtained with the Lasso: it “turns on/off” atoms in groups.

We can always consider the “noiseless” sparse coding problem $\min_{\mathbf{a}\in\mathbb{R}^p} \{\lambda\psi(\mathbf{a}) : \mathbf{x}_j = \mathbf{D}\mathbf{a}\}$, for a generic regularizer $\psi(\cdot)$, as the limit of the Lagrangian sparse coding problem $\min_{\mathbf{a}\in\mathbb{R}^p} \left\{\frac{1}{2} \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 + \lambda\psi(\mathbf{a})\right\}$ when $\lambda \to 0$. In the remainder of this section, as well as in Section III, we only present the corresponding Lagrangian formulations.

B. The Hierarchical Lasso

The Group Lasso trades sparsity at the single-coefficient level with sparsity at a group level, while, inside each group, the solution is generally dense. Let us consider for example that each group is a sub-dictionary trained to efficiently represent, via sparse modeling, an instrument or a type of image, or a given class of signals in general. The entire dictionary $\mathbf{D}$ is then appropriate to represent all classes of the signal as well as mixtures of them, and Group Lasso will properly represent sparse mixtures with

[3] While in this paper we concentrate and develop the important non-overlapping case, it will be clear that the concepts of collaborative hierarchical sparse modeling introduced here apply to the case of overlapping groups as well.

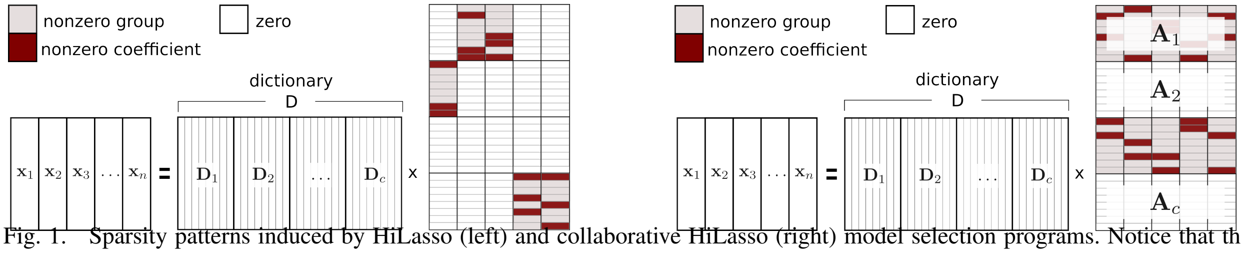


Fig. 1. Sparsity patterns induced by HiLasso (left) and collaborative HiLasso (right) model selection programs. Notice that the C-HiLasso imposes the same group-sparsity pattern in all the samples (same class), whereas the in-group sparsity patterns can vary between samples (samples themselves are different).

one group or sub-dictionary per class). At the same time, since each class is properly represented in a sparse mode via its corresponding group or sub-dictionary, we expect sparsity inside its groups as well (this is not achieved by group Lasso, whose solutions are dense inside each group). This will become even more critical in the collaborative case, where signals will share groups because they are of the same class, but will not necessarily share the full active sets, since they are not the same signal. To achieve the desired in-group sparsity, we simply re-introduce the ℓ_1 regularizer together with the group regularizer, leading to the proposed *Hierarchical Lasso (HiLasso)* model,[4]

$$\min_{\mathbf{a}\in\mathbb{R}^p} \frac{1}{2}\left\|\mathbf{x}_j - \mathbf{D}\mathbf{a}\right\|_2^2 + \lambda_2\psi_{\mathcal{G}}(\mathbf{a}) + \lambda_1\left\|\mathbf{a}\right\|_1. \qquad \text{(II.4)}$$

The hierarchical sparsity pattern produced by the solutions of (II.4) is depicted in Figure 1(left). For simplicity of the description, we assume that all the groups have the same number of elements. The extension to the general case is obtained by multiplying each group norm by the square root of the corresponding group size. This model then achieves the desired effect of promoting sparsity at the group/class level while at the same time leading to overall sparse feature selection.

The selection of λ_1 and λ_2 has an important influence on the sparsity of the obtained solution. Intuitively as λ_2/λ_1 increases, the group constraint becomes dominant and the solution tends to be more sparse at a group level but less sparse within groups (see Figure 2).

C. *Collaborative Hierarchical Lasso*

In numerous applications, one expects that certain collections of samples $\mathbf{x}_j$ share the same active components from the dictionary, that is, that the indexes of the nonzero coefficients in $\mathbf{a}_j$ are the same for all the samples in the collection. Imposing such dependency in the ℓ_1 regularized regression problem gives rise to the so called collaborative (also called “multitask” or “simultaneous”) sparse coding problem [4], [8], [9], [23].

More specifically, if we consider the matrix of coefficients $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{p\times n}$ associated with the reconstruction of the samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$, the collaborative sparse coding model is given by

$$\min_{\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{2}\left\|\mathbf{X} - \mathbf{D}\mathbf{A}\right\|_F^2 + \lambda\sum_{k=1}^{p}\left\|\mathbf{a}^k\right\|_2, \qquad \text{(II.5)}$$

where $\mathbf{a}^k \in \mathbb{R}^n$ is the k-th row of $\mathbf{A}$, that is, the vector of the n different values that the coefficient associated to the k-th atom takes for each sample $j = 1, \ldots, n$. If we now extend this idea to the Group

[4] We can similarly define a hierarchical sparsity model with ℓ_0 instead of ℓ_1.

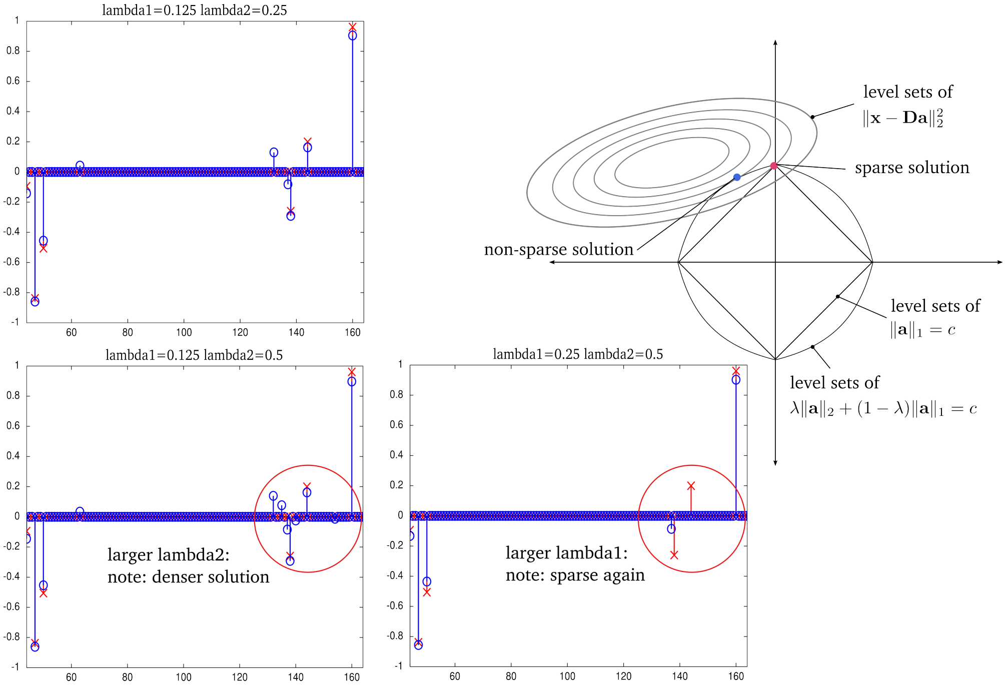


Fig. 2. Effect of different combinations of λ_1 and λ_2 on the solutions of the HiLasso coding problem. Three cases are given in which we want to recover a sparse signal (red crosses) $\mathbf{a}_0$ by means of the solution $\mathbf{a}$ of the HiLasso problem (blue dots). In this example we have two active groups out of ten possible (the sub dictionaries associated to each group have 30 atoms) and $\mathbf{a}_0 = 8$ (four non-zero coefficient per active group). The best estimate is shown in the top left. As the ratio λ_2/λ_1 increases (bottom left), the level sets of the regularizer $\psi_{\mathcal{G}}(\cdot)$ become rounder, thus encouraging denser solutions. This is depicted in the rightmost figure for a simple case of $|\mathcal{G}| = 1$. Increasing λ_1 again (bottom right) increases sparsity, although here the final effect is too strong and some non-zero coefficients are not detected.

Lasso, we obtain a *collaborative Group Lasso* (*C-GLasso*) formulation,

$$\min_{\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{2}\|\mathbf{X}-\mathbf{DA}\|_F^2 + \lambda\psi_{\mathcal{G}}(\mathbf{A}), \tag{II.6}$$

where $\psi_{\mathcal{G}}(\mathbf{A}) = \sum_{G\in\mathcal{G}} \left\|\mathbf{A}^G\right\|_F$, being $\mathbf{A}^G$ the submatrix formed by all the rows belonging to group G. This regularizer is the natural extension of the regularizer in (II.3) for the collaborative case.

In this paper we are moving one step forward and treat this together with the hierarchical extension presented in the previous section. The combined model that we propose, *C-HiLasso*, is given by

$$\min_{\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{2}\|\mathbf{X}-\mathbf{DA}\|_F^2 + \lambda_2\psi_{\mathcal{G}}(\mathbf{A}) + \sum_{j=1}^{n}\lambda_1\|\mathbf{a}_j\|_1 \,. \tag{II.7}$$

The sparsity patterns obtained with solutions to (II.7) is shown in Figure 1(right). The collaborative Group Lasso is a particular case of our model when λ_1 is zero. On the other hand, one can obtain independent Lasso for each $\mathbf{x}_i$ by setting λ_2 to zero. We see that (II.7) encourages all the signals to share the same groups (classes), while the active set inside each groups is signal dependent. We thereby obtain a collaborative hierarchical sparse model, with collaboration at the class level (all signals collaborate to identify the classes), and freedom at the individual levels inside the class to adapt to each particular signal. This new model is particularly well suited, for example, when the data vectors have missing components. In this case combining the information from all the samples is very important in order to lead to a correct representation and model (group) selection. This can be done by slightly changing the data term in (II.7). For each data vector $\mathbf{x}_j$ one computes the reconstruction error using only the observed elements. Note that the missing components do not affect the other terms of the equation. Examples will be shown in Section V.

D. Relationship to Recent Literature

A number of recent works have addressed hierarchy, grouping and collaboration within the sparse modeling community. We now discuss the most closely related to the proposed C-HiLasso model.

In [2], the authors propose a general framework in which one can define a regularization term to encourage a variety of sparsity patterns, and provide theoretical results (different than the ones developed here) for the single-signal case. The HiLasso model presented here, in the single signal scenario, can be seen as a particular case of that model (were the groups in [2] should be blocks and singletons), although the particularly and important case of hierarchical structure introduced here is not mentioned in that paper. In [15] the authors simultaneously (see [14]) proposed a model that coincides with ours again in the single-signal scenario. None of these approaches develop the collaborative framework introduced here nor the theoretical guarantees we develop.

The recovery of mixed signals with ℓ_0 optimization was addressed in [19]. This model does not include block sparsity (no hierarchy), neither collaboration. The theoretical results we obtain are not present in [19].

The special case of C-HiLasso when $\lambda_1 = 0$, which we refer to as collaborative Group Lasso (C-GLasso) here, is investigated in [24], where a theoretical analysis of the signal recovery properties of the model is developed. Collaborative coding with structured sparsity has also been used recently in the context of gene expression analysis [16], [17]. In [16], the authors propose a model, that can be interpreted as a particular case of the collaborative approach presented here, in which a set of signals is simultaneously coded using a small (sparse) number of atoms of the dictionary. They modify the classical collaborative sparse coding regularization so that each signal can use any subset of the detected atoms. This is equivalent to our model when the groups have only one element and therefore there is no hierarchy in the coding. A collaborative model is presented in [17], where signals sharing the same active atoms are grouped together in a hierarchical way by means of a tree structure. The regularization term proposed is analogous to the one proposed in our work, but it is used to group signals rather than atoms (features), having once again no hierarchical coding.

Trees have also been used recently to learn dictionaries [18]. The tree-based sparse coding is such that if a particular learned atom is not used in the decomposition of a signal, then none of its descendants (in terms of the given tree structure) can be used.

To conclude, while particular instances of the proposed C-HiLasso have been recently reported in the literature, none of them are as comprehensive. C-HiLasso includes both collaboration, at a block/group level, and hierarchical coding. Such collaborative hierarchical structure is novel and fundamental to address new important problems such as collaborative source identification and separation. The new theoretical results presented here which extend the block sparsity results of [3], [25] complement the previous modeling and algorithmic work.

III. Optimization

A. Single-Signal Problem: HiLasso

In the last decade, optimization of problems of the form of (II.2) and (II.3) have been deeply studied, and there exist very efficient algorithms for solving them. Recently, Wright et. al [10] proposed a framework, SpaRSA, for solving the general problem

$$\min_{\mathbf{a}\in\mathbb{R}^p} f(\mathbf{a}) + \lambda\psi(\mathbf{a}). \tag{III.8}$$

To guarantee convergence, $f : \mathbb{R}^p \to \mathbb{R}$ needs to be a smooth and convex function, while $\psi : \mathbb{R}^p \to \mathbb{R}$ only needs to be finite in $\mathbb{R}^p$. This formulation includes as important particular cases the Lasso, Group-Lasso and HiLasso problems by setting $f(\cdot)$ as the reconstruction error and then choosing the corresponding regularizers for $\psi(\cdot)$. When the regularizer, $\psi(\cdot)$, is group separable, the optimization can be subdivided into smaller problems, one per group. The framework becomes powerful when these subproblems can be solved efficiently. This is the case of the Lasso and Group Lasso (with non overlapping groups) settings but is not immediate with the HiLasso regularizer (II.4). In this work we combine SpaRSA with the ADMOM method [11] to efficiently solve the HiLasso problem.

The SpaRSA algorithm generates a sequence of iterates $\{\mathbf{a}^{(t)}\}_{t\in\mathbb{N}}$ that, under certain conditions, converges to the solution of (III.8). At each iteration, $\mathbf{a}^{(t+1)}$ is obtained by solving

$$\min_{\mathbf{z}\in\mathbb{R}^p} (\mathbf{z}-\mathbf{a}^{(t)})^T\nabla f(\mathbf{a}^{(t)}) + \frac{\alpha^{(t)}}{2}\left\|\mathbf{z}-\mathbf{a}^{(t)}\right\|_2^2 + \lambda\psi(\mathbf{z}), \tag{III.9}$$

for some sequence of parameters $\{\alpha^{(t)}\}_{t\in\mathbb{N}}$, $\alpha^{(t)} \in \mathbb{R}^+$, which needs to be chosen properly for the algorithm to converge (see [10] for details). It is easy to show that (III.9) is equivalent to

$$\min_{\mathbf{z}\in\mathbb{R}^p} \frac{1}{2}\left\|\mathbf{z}-\mathbf{u}^{(t)}\right\|_2^2 + \frac{\lambda}{\alpha^{(t)}}\psi(\mathbf{z}), \tag{III.10}$$

where $\mathbf{u}^{(t)} = \mathbf{a}^{(t)} - \frac{1}{\alpha^{(t)}}\nabla f(\mathbf{a}^{(t)})$. In this new formulation, it is clear that the first term in the cost function can be separated element-wise. Thus, when the regularization function $\psi(\mathbf{z})$ is group separable, so is the overall optimization, and one can solve (III.10) independently for each group, leading to

$$\min_{\mathbf{z}_G\in\mathbb{R}^{|G|}} \frac{1}{2}\left\|\mathbf{z}_G-\mathbf{u}_G^{(t)}\right\|_2^2 + \frac{\lambda}{\alpha^{(t)}}\psi_G(\mathbf{z}_G),$$

$\mathbf{z}_G$ being the corresponding variable for the group. In the case of HiLasso, this becomes,

$$\min_{\mathbf{b}\in\mathbb{R}^{|G|}} \frac{1}{2}\left\|\mathbf{b}-\mathbf{w}\right\|_2^2 + \frac{\lambda_2}{\alpha^{(t)}}\left\|\mathbf{b}\right\|_2 + \frac{\lambda_1}{\alpha^{(t)}}\left\|\mathbf{b}\right\|_1, \tag{III.11}$$

where $\mathbf{w} = \mathbf{u}_G^{(t)}$ is the subvector of $\mathbf{u}^{(t)} = \mathbf{a}^{(t)} - \frac{1}{\alpha^{(t)}}\mathbf{D}^{(t)}(\mathbf{D}\mathbf{a}^{(t)} - \mathbf{x})$ indexed by G. Problem (III.11) is a second order cone programing (SOCP), for which one could use generic solvers. However, this subproblem needs to be solved many times within the SpaRSA iterations, so it is crucial to solve it efficiently.

To obtain an efficient implementation of SpaRSA, we use the ADMOM method [11]. The idea is to solve the artificially constrained equivalent problem,

$$\min_{b\in\mathbb{R}^{|G|}} \frac{1}{2}\left\|\mathbf{b}-\mathbf{w}\right\|_2^2 + \tilde{\lambda}_2\left\|\beta\right\|_2 + \tilde{\lambda}_1\left\|\mathbf{b}\right\|_1, \quad \text{s.t.} \quad \mathbf{b}=\beta,$$

where $\tilde{\lambda}_i = \lambda_i/\alpha^{(t)}$. The algorithm generates a set of iterates $\{\mathbf{b}^{(t)}, \beta^{(t)}, \mathbf{p}^{(t)}\}_{t\in\mathbb{N}^+}$ which converges to the minimum of the Augmented Lagrangian

$$L_c(\mathbf{b},\beta,\mathbf{p}) = \frac{1}{2}\left\|\mathbf{b}-\mathbf{w}\right\|_2^2 + \tilde{\lambda}_2\left\|\beta\right\|_2 + \tilde{\lambda}_1\left\|\mathbf{b}\right\|_1 + \mathbf{p}^T(\mathbf{b}-\beta) + \frac{c}{2}\left\|\mathbf{b}-\beta\right\|_2^2.$$

The elements of $\mathbf{p}$ are the so called Lagrangian multipliers, and $c > 0$ is the augmented Lagrangian penalty term, which is a parameter of the algorithm. The idea of adding a quadratic term $\frac{c}{2}\left\|\mathbf{b}-\beta\right\|_2^2$ to the Lagrangian is to render the corresponding primal cost function strictly convex, so that the dual function is differentiable. This in turn allows the optimization to be carried out using primal-dual iterations, where the dual variables, in this case $\mathbf{p}$, can be updated efficiently using gradient ascent. The method always converges to the optimum, and the parameter c needs to be chosen empirically in order to obtain a good convergence rate.

Each iteration of the ADMOM method updates $\mathbf{b}$, β and $\mathbf{p}$, one at a time, while leaving the others fixed. The updates of $\mathbf{b}$ and β are obtained via exact minimization of the augmented Lagrangian in $\mathbf{b}$ and β respectively, while the update of the Lagrangian multiplier $\mathbf{p}$ is a gradient ascent towards the solution of the dual problem. This leads to the iterations:

$$\begin{aligned}\mathbf{b}^{(t+1)} =& \underset{\mathbf{b}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{b}-\mathbf{w}\right\|_2^2 + \tilde{\lambda}_1\left\|\mathbf{b}\right\|_1 + \mathbf{b}^T\mathbf{p} + \frac{c}{2}\left\|\mathbf{b}-\beta\right\|_2^2, \qquad \text{(III.12)}\\ \beta^{(t+1)} =& \underset{\beta}{\operatorname{argmin}}\ \tilde{\lambda}_2\left\|\beta\right\|_2 - \beta^T\mathbf{p} + \frac{c}{2}\left\|\mathbf{b}^{(t+1)}-\beta\right\|_2^2,\\ \mathbf{p}^{(t+1)} =& \mathbf{p} + c(\mathbf{b}^{(t+1)} - \beta^{(t+1)}).\end{aligned}$$

For convenience of notation we omitted the super-indexes for the iterates at step t, just explicitly indexing them at step $t+1$. For more details on augmented Lagrangian methods see [11, Chapter 3].

The update for $\mathbf{b}$ is separable into scalar subproblems on the coordinates of $\mathbf{b}$. The optimality conditions on the subgradient of each of these scalar problems leads to a simple variant of the well known soft-thresholding operator, $\mathcal{S}(w_i,\lambda) = \operatorname{sgn}(w_i)\max\{0,|w_i|-\lambda\}$. We use $\mathcal{S}(\mathbf{w},\lambda)$ to denote the vector obtained when applying the soft-thresholding operator (with parameter λ) to each element of $\mathbf{w}$. On the other hand, the update for β is not separable into scalar subproblems. However, we show now that its solution can be obtained in closed form via vector shrinkage. For this, we write the optimality conditions on the subgradient of (III.12),

$$\tilde{\lambda}_2\partial\left\|\beta\right\|_2 - \mathbf{p} - c(\mathbf{b}-\beta) \ni \mathbf{0} \Rightarrow c\beta + \tilde{\lambda}_2\partial\left\|\beta\right\|_2 - \mathbf{p} - c\mathbf{b} \ni \mathbf{0},$$

and perform the change of variables $\beta' = c\beta$ and $\mathbf{b}' = \mathbf{p} + c\mathbf{b}$. Since the subgradient of $\left\|\mathbf{b}\right\|_2$ is not modified by a constant scaling of the argument, we obtain an equivalent optimality condition, $\beta' + \tilde{\lambda}_2\partial\left\|\beta'\right\|_2 - \mathbf{b}' \ni \mathbf{0}$, which is exactly the one leading to the vector shrinkage operator, $\mathcal{S}_v$ described in [1] for the Group Lasso (actually much simpler, since there is no matrix multiplication involved), $\mathcal{S}_v(\mathbf{b}',\tilde{\lambda}_2) = \left[1 - \frac{\tilde{\lambda}_2}{\left\|\mathbf{b}'\right\|_2}\right]_+\mathbf{b}'$. After reverting the change of variables, we get closed form updates for $\mathbf{b}$ and β,

$$\mathbf{b} = \frac{1}{c+1}\mathcal{S}(\mathbf{w}+c\beta-\mathbf{p},\tilde{\lambda}_1) \quad \text{and} \quad \beta = \frac{1}{c}\,\mathcal{S}_v(\mathbf{p}+c\mathbf{b},\tilde{\lambda}_2).$$

The complete HiLasso optimization algorithm is summarized in Algorithm 1. The parameter η is needed to guarantee the convergence of SpaRSA and has very little influence in the overall performance (see [10] for details), we used $\eta = 2$ in all our experiments. An additional speed up is obtained by bypassing ADMOM when a whole group is not active. From the optimality conditions of (III.11), it follows that if $\mathbf{0}$ is a solution when $\lambda_1 = 0$ (standard Group Lasso) or $\lambda_2 = 0$ (Lasso), it is also a solution in the general case. This can be simply checked by evaluating $\mathcal{S}_v(\,\mathbf{w}\,,\tilde{\lambda}_2) > \mathbf{0}$ and $\mathcal{S}(\,\mathbf{w}\,,\tilde{\lambda}_1) > \mathbf{0}$ before proceeding with the ADMOM algorithm.

```
Input: Data $\mathbf{X}$, dictionary $\mathbf{D}$, group set $\mathcal{G}$, constants $\eta > 1$, $c > 0$, $0 < \alpha_{\text{min}} < \alpha_{\text{max}}$
Output: The optimal point $\mathbf{a}^*$
Initialize $t := 0, \mathbf{a}(0) := \mathbf{0}$;
while stopping criterion is not satisfied do
    choose $\alpha^{(t)} \in [\alpha_{\text{min}}, \alpha_{\text{max}}]$;
    set $\mathbf{u}^{(t)} := \mathbf{a}^{(t)} - \frac{1}{\alpha^{(t)}}\nabla f(\mathbf{a}^{(t)})$;
    while stopping criterion is not satisfied do
        // Here we use the group separability of (III.10) and solve (III.11) for each group
        for $i := 1$ to $|\mathcal{G}|$ do
            if $\mathcal{S}_v(\mathbf{w}, \tilde{\lambda}_2) > \mathbf{0}$ then
                set $r := 0$;
                choose an initial $\mathbf{p}^0, \beta^0, \mathbf{b}^0$;
                while stopping criterion is not satisfied do
                    $\mathbf{b}^{(r+1)} = \frac{1}{c+1}\mathcal{S}(\mathbf{u_i^{(t)}} + c\beta^{(r)} - \mathbf{p^{(r)}}, \tilde{\lambda}_1)$;
                    $\beta^{(r+1)} = \frac{1}{c}\mathcal{S}_v(\mathbf{p}^{(r)} + c\,\mathbf{b}^{(r+1)}, \tilde{\lambda}_2)$;
                    $\mathbf{p}^{(r+1)} = \mathbf{p}^{(r)} + c(\mathbf{b}^{(r+1)} - \beta^{(r+1)})$;
                    set $r := r + 1$ ;
                end
                set $\mathbf{a}_G^{(t+1)} := \mathbf{b}^{r+1}$ ;
            else
                set $\mathbf{a}_G^{(t+1)} := \mathbf{0}$;
            end
        end
        set $\alpha^{(t)} := \eta\alpha^{(t)}$;
    end
    set $t := t + 1$ ;
end
```

Algorithm 1: HiLasso optimization algorithm.

It is interesting to note that when either $\lambda_1 = 0$ or $\lambda_2 = 0$, the ADMOM technique is not needed and the SpaRSA subproblem (III.11) is solved exactly in one step using either a soft thresholding (when $\lambda_2 = 0$) or a vector thresholding (when $\lambda_1 = 0$). In particular, when $\lambda_2 = 0$, the proposed optimization then reduces to the Iterative Soft Thresholding algorithm [26].

B. Optimization of the Collaborative HiLasso

We now propose an optimization algorithm to efficiently solve the collaborative HiLasso. The main idea is to include a second use of ADMOM in order to divide the overall problem into two subproblems: one that breaks the multi-signal problem into n single-signal ℓ_1-based sparse codings, and another that treats the multi-signal case as a single Group Lasso-like problem. In this way we take advantage of the separability of each term.

We define a constrained optimization problem,

$$\min_{\mathbf{A},\mathbf{B}\in\mathbb{R}^{p\times n}} \frac{1}{2}\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F^2 + \lambda_1\sum_j \|\mathbf{a}_j\|_1 + \lambda_2\psi_{\mathcal{G}}(\mathbf{B}) \text{ s.t. } \mathbf{A}=\mathbf{B}.$$

The ADMOM iterations are given by (we omitted the super-index for variables at iteration t for notational convenience).

$$\mathbf{A}^{(t+1)} = \underset{\mathbf{A}}{\mathrm{argmin}} \frac{1}{2} \left\|\mathbf{X}-\mathbf{D}\mathbf{A}\right\|_F^2 + \lambda_1 \sum_j \left\|\mathbf{a}_j\right\|_1 + \mathrm{Tr}(\mathbf{A}^T\mathbf{P}) + \frac{c}{2}\left\|\mathbf{B}-\mathbf{A}\right\|_F^2, \tag{III.13}$$

$$\mathbf{B}^{(t+1)} = \underset{\mathbf{B}}{\mathrm{argmin}} \frac{c}{2} \left\|\mathbf{B}-\mathbf{A}^{(t+1)}\right\|_F^2 + \mathrm{Tr}(\mathbf{B}^T\mathbf{P}) + \lambda_2 \psi_{\mathcal{G}}(\mathbf{B}), \tag{III.14}$$

$$\mathbf{P}^{(t+1)} = \mathbf{P} + c(\mathbf{A}^{(t+1)} - \mathbf{B}^{(t+1)}).$$

Solving for $\mathbf{A}^{(t+1)}$: Problem (III.13) can be separated into n single-signal subproblems by updating one column of the matrix $\mathbf{A}$ at a time,

$$\min_{\mathbf{a}_j \in \mathbb{R}^p} \frac{1}{2}\left\|\mathbf{x} - \mathbf{D}\mathbf{a}_j\right\|_2^2 + \mathbf{p}_j^T\mathbf{a}_j + \frac{c}{2}\left\|\mathbf{a}_j - \mathbf{b}\right\|_2^2 + \lambda_1 \left\|\mathbf{a}_j\right\|_1.$$

This problem can be solved using the SpaRSA framework following the ideas used in the previous section. In this case the function f on (III.8) would be the first three terms on the equation above and ψ is the standard ℓ_1 regularization term. The idea is to consider the first three terms of the cost as $f(\cdot)$ in Equation (III.8). The associated computational cost is equivalent to the one of the Lasso, since the regularizer is the standard ℓ_1 norm.

Solving for $\mathbf{B}^{(t+1)}$: The problem given by (III.14) is group separable, as a direct consequence of the separability of $\psi_{\mathcal{G}}$ for the case of non overlapping groups. Thus, we need to solve $|\mathcal{G}|$ optimization problems of the form,

$$\min_{\mathbf{B}_G \in \mathbb{R}^{g\times n}} \frac{c}{2}\left\|\mathbf{B}_G - \mathbf{A}_G^{(t+1)}\right\|_F^2 + \mathrm{Tr}(\mathbf{P}_G^{(t+1)}\mathbf{B}_G^T) + \lambda_2 \left\|\mathbf{B}_G\right\|_F,$$

where $\mathbf{A}_G$, $\mathbf{B}_G$ and $\mathbf{P}_G$ are the $|G| \times n$ sub-matrices of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{P}$ associated with the group G respectively. We express them as column vectors (each with $|G|n$ components) by concatenating their columns, obtaining $\mathbf{b}_G, \mathbf{a}_G$ and $\mathbf{p}_G$ respectively, and rewrite the optimization problem in vectorial form as

$$\min_{\mathbf{b} \in \mathbb{R}^{|\mathbf{G}|\mathbf{n}}} \lambda_2 \left\|\mathbf{b}\right\|_2 - \mathbf{p}_G^T\mathbf{b} + \frac{c}{2}\left\|\mathbf{a}_G^{(t+1)} - \mathbf{b}\right\|_2^2.$$

This problem is identical to (III.12) and can be reduced to a Group Lasso problem by simply changing variables and thus, it is solved using vector thresholding.

Similarly to what we obtained in the single signal case, the developed optimization procedure alternates between a Lasso-like problem (solved by an iterative shrinkage procedure, via SpaRSA) applied to each signal independently, and a vector soft-thresholding at a group level. The important difference here is that the group thresholding is applied considering all the signals. Intuitively this means that for a group to be treated as active, the atoms of its corresponding subdictionary need to be relevant for all (or a significant number of) the signals in the set, which translates into robustness in the model (class) selection.

As with models such as Lasso and Group Lasso, the optimal parameters λ_1 and λ_2 are application and data dependent. In some specific cases, closed form solutions exist for such parameters. For example, for signal restoration in the presence of noise and using Lasso ($\lambda_2 = 0$), the GSURE method gives a closed form solution for λ_1 [27]. As extending such methods to C-HiLasso is beyond the scope of this work, we rely on cross-validation for the choice of such parameters. Our optimization technique also relies on the choice of the augmented Lagrangian quadratic penalty c. Although there is no closed form solution for an optimal value of c, in practice the performance of the algorithms is very robust to its choice, and a small coarse grid is enough to choose a single suitable value that works well for the range of problems in Section V.

IV. Theoretical guarantees

In our current theoretical analysis, we analyze the case of a single measurement vector (signal) $\mathbf{x}$ (we comment on the collaborative case at the end of this section), and assume that there is no measurement noise or perturbation, so that $\mathbf{x} = \mathbf{D}\mathbf{a}$. We attempt to recover $\mathbf{a}$ by solving the noise-free HiLasso problem:

$$\min_{\mathbf{a} \in \mathbb{R}^p} \left\{ \lambda \psi_{\mathcal{G}}(\mathbf{a}) + (1-\lambda) \|\mathbf{a}\|_1 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\mathbf{a} \right\}. \tag{IV.15}$$

Note that we have replaced the two regularization parameters λ_1 and λ_2 by a single parameter λ, since scaling does not effect the optimal solution. We can always assume that $\lambda_1 + \lambda_2 = 1$.

Our goal is to develop guarantees under which the HiLasso program of (IV.15) will recover the true unknown vector $\mathbf{a}$. We assume throughout this section that $\mathbf{a}$ has group sparsity k, namely, not more than k of the group vectors $\mathbf{a}_G$, $G \in \mathcal{G}$, have non-zero norm. In addition, within each group, we assume that not more than s elements are non zero, that is, $\|\mathbf{a}_G\|_0 \leq s$. Without loss of generality, we further assume that the length of each vector $\mathbf{a}_G$ is equal to g.

For $\lambda = 1$ the problem (IV.15) reduces to the mixed ℓ_2/ℓ_1 problem formally studied in [3], [25]. When $\lambda = 0$, (IV.15) becomes equivalent to the well-known Lasso, or basis pursuit algorithm. Both cases have been treated previously in the literature and sufficient conditions have been derived on the sparsity levels and on the dictionary $\mathbf{D}$ to ensure that the resulting optimization problem recovers the true unknown vector. For example, in [3], [28], [29] conditions are given in terms of the restricted isometry property (RIP) of $\mathbf{D}$. An alternative line of work [25], [30], focused on coherence guarantees, which are easier to compute. Here, we follow the same spirit and consider coherence bounds that ensure recovery using the HiLasso approach. We also draw from [9] to briefly describe conditions under which the probability of error of recovering the correct groups, using the special case of the C-HiLasso with $\lambda_1 = 0$ (C-GLasso), falls exponentially to 0 as the number of collaborating samples n grows.

A. Block-Sparse Coherence

We begin by reviewing previously proposed coherence measures. For a given dictionary $\mathbf{D}$, the (standard) coherence is defined as $\mu = \max_{i,j \neq i} |\mathbf{d}_i^T \mathbf{d}_j|$, where $\mathbf{d}_i$ denotes the ith column of the dictionary $\mathbf{D}$. This coherence was extended to the block-sparse setting in [25], leading to the definition of block coherence:[5]

$$\mu_B = \max_{i,j \neq i} \frac{1}{g} \rho(\mathbf{D}_i^T \mathbf{D}_j),$$

where $\rho(\cdot)$ is the spectral norm, that is, $\rho(\mathbf{Z}) = \lambda_{\max}^{1/2}(\mathbf{Z}^T\mathbf{Z})$ with $\lambda_{\max}(\mathbf{W})$ denoting the largest eigenvalue of the positive semi-definite matrix $\mathbf{W}$. When $g = 1$ (each block is a singleton), $\mathbf{D}_i = \mathbf{d}_i$, so that as expected, $\mu_B = \mu$. While μ_B quantifies global properties of the dictionary $\mathbf{D}$, local properties are characterized by the sub-coherence of $\mathbf{D}$, defined as

$$\nu = \max_{G \in \mathcal{G}} \left\{ \max_{i,j \neq i} |\mathbf{d}_i^T \mathbf{d}_j|, \quad \mathbf{d}_i, \mathbf{d}_j \in \mathbf{D}_G \right\}. \tag{IV.16}$$

We define $\nu = 0$ for $g = 1$. Clearly, if the columns of $\mathbf{D}_G$ are orthonormal for each group G, then $\nu = 0$. Assuming the columns of $\mathbf{D}$ have unit norm, it can be easily shown that μ, ν and μ_B all lie in the range $[0, 1]$. In addition, we can easily prove that $\nu \leq \mu$ and $\mu_B \leq \mu$. In our setting $\mathbf{a}$ is block sparse, but has further internal structure: each subvector of $\mathbf{a}$ is also sparse. Therefore, we expect that an appropriate

[5] Each sub-dictionary $\mathbf{D}_i$ corresponds to one of the (non-overlapping) groups in $\mathcal{G}$ considered in HiLasso.

coherence measure will be based on the definition of block sparsity, but will further incorporate the internal sparsity. Let $\mathbf{M} = \mathbf{D}^T\mathbf{D}$ denote the Gram matrix of inner products of the column of $\mathbf{D}$. Then, the standard block coherence μ is defined in terms of the largest singular value of an off-diagonal sub-block of $\mathbf{M}$. In a similar fashion, we will define sparse block coherence in terms of sparse singular values. As we will see, two different definitions will play a role, depending on where exactly the sparsity within the block enters.

To define the sparse block coherence measures, we note that the spectral norm $\rho(\mathbf{Z})$ of a matrix $\mathbf{Z}$ can be defined as

$$\rho(\mathbf{Z}) = \max_{\mathbf{x},\mathbf{y}} |\mathbf{x}^T\mathbf{Z}\mathbf{y}| \qquad \text{s.t.} \quad \|\mathbf{x}\| = 1, \|\mathbf{y}\| = 1.$$

Alternatively, we can define $\rho(\mathbf{Z})$ as above, via the largest eigenvalue $\lambda_{\max}$ of $\mathbf{W} = \mathbf{Z}^T\mathbf{Z}$: $\rho(\mathbf{Z}) = \sqrt{\lambda_{\max}(\mathbf{W})}$. Formally, for any $\mathbf{W} \succeq 0$,

$$\lambda_{\max}(\mathbf{W}) = \max_{\mathbf{y}} \mathbf{y}^T\mathbf{W}\mathbf{y} \qquad \text{s.t.} \quad |\mathbf{y}\| = 1,$$

We now develop sparse analogs of $\rho(\mathbf{Z})$ and $\lambda_{\max}(\mathbf{Z}^T\mathbf{Z})$. As we will see, the simple square-root relation no longer holds in this case. To define the sparse largest singular value, we restrict $\mathbf{x}$ and $\mathbf{y}$ to be s-sparse [31]:

$$\rho^{ss}(\mathbf{Z}) = \max_{\mathbf{x},\mathbf{y}} |\mathbf{x}^T\mathbf{Z}\mathbf{y}| \qquad \text{s.t.} \quad \|\mathbf{x}\| = 1, \|\mathbf{y}\| = 1, \|\mathbf{x}\|_0 \leq s, \|\mathbf{y}\|_0 \leq s. \tag{IV.17}$$

Similarly, the largest sparse eigenvalue of $\mathbf{W} = \mathbf{Z}^T\mathbf{Z}$ is defined as [31], [32], [33]

$$\lambda^s_{\max}(\mathbf{W}) = \max_{\mathbf{y}} \mathbf{y}^T\mathbf{W}\mathbf{y} \qquad \text{s.t.} \quad \|\mathbf{y}\| = 1, \|\mathbf{y}\|_0 \leq s. \tag{IV.18}$$

The sparse matrix norm is then given by

$$\rho^s(\mathbf{Z}) = \sqrt{\lambda^s_{\max}(\mathbf{Z}^T\mathbf{Z})}. \tag{IV.19}$$

Note that in general $\rho^s(\mathbf{Z})$ is not equal to $\rho^{ss}(\mathbf{Z})$. It is easy to see that $\rho^{ss}(\mathbf{Z}) \leq \rho^s(\mathbf{Z})$. Efficient algorithms for computing $\rho^s(\mathbf{Z})$ (also referred to in the literature as sparse PCA) and $\rho^{ss}(\mathbf{Z})$ can be found in [31], [32], [33]. For any matrix $\mathbf{Z}$, $\rho^{ss}(\mathbf{Z}) = \rho(\tilde{\mathbf{I}}^T\mathbf{Z}\tilde{\mathbf{I}})$ and $\rho^s(\mathbf{Z}) = \rho(\mathbf{Z}\tilde{\mathbf{I}})$, where $\tilde{\mathbf{I}}$ is a matrix with s columns, which consist of s nonzero rows and all remaining rows identically zero. The nonzero rows contain only one nonzero element, taking on the value 1. The location of the ones are chosen to maximize the corresponding singular value. In the definition of ρ^{ss}, the two matrices $\tilde{\mathbf{I}}$ do not necessarily correspond to the same support selection. However, in order to not further complicate the notation, we denote both matrices by $\tilde{\mathbf{I}}$.

Using (IV.17) and (IV.19), we define two sparse block coherence measures:

$${\mu_B}^{ss} = \max_{i,j\neq i} \frac{1}{g}\rho^{ss}(\mathbf{D}_i^T\mathbf{D}_j), \tag{IV.20}$$

and

$${\mu_B}^{s} = \max_{i,j\neq i} \frac{1}{g}\rho^{s}(\mathbf{D}_i^T\mathbf{D}_j). \tag{IV.21}$$

The choice of scaling is to ensure that ${\mu_B}^s, {\mu_B}^{ss} \leq \mu_B$. We also extend the notion of sub-coherence to account for the internal sparsity. Specifically, we define

$$\nu^s = \max_{i} \frac{1}{g}\rho(\tilde{\mathbf{I}}^T\mathbf{D}_i^T\overline{\mathbf{D}_i}), \tag{IV.22}$$

where $\overline{\mathbf{D}_i}$ denotes the $g - s$ columns of $\mathbf{D}_i$ not selected by $\tilde{\mathbf{I}}$. In other words, ν^s measures the coherence between the active part of each block and the non-active section.

The following proposition establishes some relations between these new definitions and the standard coherence measures.

Proposition 1. The sparse block-coherence measures ${\mu_B}^{ss}, {\mu_B}^s$ satisfy

$$0 \le {\mu_B}^{ss} \le \frac{s}{g}\mu, \quad 0 \le {\mu_B}^s \le \sqrt{\frac{s}{g}}\mu. \tag{IV.23}$$

The sparse sub-coherence satisfies

$$0 \le \nu^s \le \frac{\sqrt{s(g-s)}}{g}\nu. \tag{IV.24}$$

Proof: The inequalities ${\mu_B}^{ss}, {\mu_B}^s \ge 0$ follow immediately from the definition. We obtain the upper bounds by rewriting $\rho^{ss}(\mathbf{Z})$ and $\rho^s(\mathbf{Z})$ and then using the Guers̆gorin disc theorem,

$$\rho^{ss}(\mathbf{Z}) \;=\; \lambda^{1/2}_{\max}(\tilde{\mathbf{I}}^T\mathbf{Z}^T\tilde{\mathbf{I}}\tilde{\mathbf{I}}^T\mathbf{Z}\tilde{\mathbf{I}}) \overset{(a)}{\le} \sqrt{\max_l \sum_{r=1}^{s} |e_{lr}|} \le \sqrt{s \max_{l,r} |e_{lr}|} \tag{IV.25}$$

$$\rho^{s}(\mathbf{Z}) \;=\; \lambda^{1/2}_{\max}(\tilde{\mathbf{I}}^T\mathbf{Z}^T\mathbf{Z}\tilde{\mathbf{I}}) \overset{(b)}{\le} \sqrt{\max_l \sum_{r=1}^{s} |e'_{lr}|} \le \sqrt{s \max_{l,r} |e'_{lr}|} \tag{IV.26}$$

where e_{lr} and e'_{lr} are the elements of $\mathbf{E} = \mathbf{M}_{ij}^T\tilde{\mathbf{I}}\tilde{\mathbf{I}}^T\mathbf{M}_{ij}$ and $\mathbf{E}' = \mathbf{M}_{ij}^T\mathbf{M}_{ij}$, and (a), (b) are a consequence of Gers̆gorin's disc theorem. The entries of $\mathbf{M}_{ij} = \mathbf{D}_i^T\mathbf{D}_j$ for $i \ne j$ have absolute value smaller than or equal to μ, and the size of $\mathbf{M}_{ij}$ is $g \times g$. Therefore, $|e_{k\ell}| \le s\mu^2$ and $|e'_{k\ell}| \le g\mu^2$. Substituting these values into (IV.25) and (IV.26) concludes the proof of the upper bounds on ${\mu_B}^{ss}$ and ${\mu_B}^s$.

The proof of ν^s follows a similar path where now the size of $\mathbf{M}_{ij} = \mathbf{D}_i^T\overline{\mathbf{D}_i}$ is $g \times (g - s)$. ■

B. Recovery Proof

To formally state our main recovery result, suppose that $\mathbf{a}_0$ is a block k-sparse vector with blocks of length g, where each block has sparsity s, and let $\mathbf{x} = \mathbf{D}\mathbf{a}_0$. Our theorem relies on the following definitions, which we will use throughout this section. Let $\mathbf{D}_0$ denote the matrix whose blocks correspond to the nonzero blocks of $\mathbf{a}_0$, and let $\mathbf{c}_0$ be the corresponding coefficient blocks. Denote by $\mathbf{c}_0^S$ the nonzero elements of $\mathbf{c}_0$ (namely, the nonzero values in each block), and similarly let $\mathbf{D}_0^S$ indicate the respective matrices on the support S so that $\mathbf{x} = \mathbf{D}_0\mathbf{c}_0 = \mathbf{D}_0^S\mathbf{c}_0^S$. We let $\overline{\mathbf{D}}_0$ be the matrix which contains the blocks of size g of $\mathbf{D}$ that are not in $\mathbf{D}_0$, and let $\mathbf{D}_0^{\overline{S}}$ contain the columns of $\mathbf{D}_0$ that are not in $\mathbf{D}_0^S$. Finally, we define $\overline{\mathbf{D}}$ as the matrix containing all columns not in $\mathbf{D}_0^S$. In terms of our other definitions, $\overline{\mathbf{D}}$ is a concatenation of $\overline{\mathbf{D}}_0$ and $\mathbf{D}_0^{\overline{S}}$.

An important observation that we will rely on throughout, is that the columns of $\mathbf{D}_0^S$ must be linearly independent for any choice of $\mathbf{D}_0$ and S in order to guarantee a unique sparse representation in our problem. Otherwise, we can have two sparse vectors $\mathbf{c}_0^S$ and $\mathbf{c}^S$ that satisfy $\mathbf{D}_0^S\mathbf{c}_0^S = \mathbf{D}_0^S\mathbf{c}^S$ and no algorithm will be able to distinguish between them. Under this assumption, $(\mathbf{D}_0^S)^T\mathbf{D}_0^S$ is invertible and we can define the pseudo-inverse $(\mathbf{D}_0^S)^\dagger = ((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1}(\mathbf{D}_0^S)^T$. Equipped with these definitions we can now state our main result.

Theorem 1. *Let $\mathbf{a}_0$ be a block k-sparse vector with blocks of length g, where each block has sparsity s. Let $\mathbf{x} = \mathbf{D}\mathbf{a}_0$ for a given matrix $\mathbf{D}$. A sufficient condition for the HiLasso algorithm (IV.15) to recover $\mathbf{a}_0$ is that*

$$\rho_c((\mathbf{D}_0^S)^\dagger \overline{\mathbf{D}}_0) \; < \; 1 \tag{IV.27}$$

$$\rho_c((\mathbf{D}_0^S)^\dagger \mathbf{D}_0^{\overline{S}}) \; < \; 1 \tag{IV.28}$$

$$\|(\mathbf{D}_0^S)^\dagger \overline{\mathbf{D}}\|_{1,1} \; < \; 1. \tag{IV.29}$$

Here $\rho_c(\mathbf{Z}) = \max_r \sum_\ell \rho(\mathbf{Z}_{\ell r})$, $\mathbf{Z}_{\ell r}$ denoting the (ℓ, r)th $s \times p$ block of $\mathbf{Z}$ where $p = g$ in (IV.27) and $p = g - s$ in (IV.28), and $\|\mathbf{Z}\|_{1,1} = \max_r \|\mathbf{z}_r\|_1$, where $\mathbf{z}_r$ is the rth column of $\mathbf{Z}$.

Note that (IV.27) and (IV.28) imply that for all r, $\rho_c(\mathbf{D}_0^\dagger[\overline{\mathbf{D}}_0]_r) < 1$ and $\rho_c(\mathbf{D}_0^\dagger[\mathbf{D}_0^{\overline{S}}]_r) < 1$. From (IV.29) we have that $\|(\mathbf{D}_0^S)^\dagger \mathbf{d}_r\|_1 < 1$, for every column $\mathbf{d}_r$ of $\overline{\mathbf{D}}$. On the other hand, when $\mathbf{d}_r$ is a column from $\mathbf{D}_0^S$, $(\mathbf{D}_0^S)^\dagger \mathbf{d}_r$ is a vector containing the value 1 in the location corresponding to $\mathbf{d}_r$, and zero everywhere else, thus $\|(\mathbf{D}_0^S)^\dagger \mathbf{d}_r\|_1 = 1$. Combining the last two observations we conclude that

$$\|(\mathbf{D}_0^S)^\dagger \mathbf{d}_r\|_1 \leq 1, \; \forall \, r \tag{IV.30}$$

We will use this result in the proof of the theorem.

The sufficient conditions (IV.27)–(IV.29) depend on $\mathbf{D}_0^S$ and therefore on the nonzero blocks in $\mathbf{c}_0$, and the nonzero locations within the blocks, which, of course, are not known in advance. Nonetheless, below we will prove sufficient conditions that ensure that (IV.27)–(IV.29) holds, which depend only on ${\mu_B}^{ss}, {\mu_B}^s, \nu^s$ and ν, associated with the dictionary $\mathbf{D}$.

We now prove Theorem 1.

Proof: To prove that (IV.15) recovers the correct vector $\mathbf{a}_0$, let $\mathbf{a}'$ be an alternative solution satisfying $\mathbf{x} = \mathbf{D}\mathbf{a}'$. Denote by $\mathbf{c}_0$ the blocks consisting of the nonzero values of $\mathbf{a}_0$ and by $\mathbf{D}_0$ the corresponding columns of $\mathbf{D}$. Similarly, let $\mathbf{c}'$ denote the blocks consisting of the nonzero values of $\mathbf{a}'$ and let $\mathbf{D}'$ denote the corresponding columns of $\mathbf{D}$. Then $\mathbf{x} = \mathbf{D}_0\mathbf{c}_0 = \mathbf{D}'\mathbf{c}'$.

In each block of $\mathbf{c}_0$ there are at most s nonzero values. We denote by $\mathbf{c}_0^S$ the nonzero elements of $\mathbf{c}_0$, and similarly let $\mathbf{D}_0^S$ indicate the respective matrices on the support S. Relying on the fact that $\mathbf{D}_0^S$ has linearly independent columns, we can write

$$\mathbf{c}_0^S = (\mathbf{D}_0^S)^\dagger \mathbf{D}_0^S \mathbf{c}_0^S = \mathbf{Q}\mathbf{D}_0^S \mathbf{c}_0^S, \tag{IV.31}$$

where we denoted $\mathbf{Q} = (\mathbf{D}_0^S)^\dagger$ for brevity. Noting that $\mathbf{D}_0^S \mathbf{c}_0^S = \mathbf{D}_0\mathbf{c}_0 = \mathbf{D}'\mathbf{c}'$, (IV.31) becomes

$$\mathbf{c}_0^S = \mathbf{Q}\mathbf{D}'\mathbf{c}'. \tag{IV.32}$$

To proceed, we separate $\mathbf{D}'$ into two parts: blocks that are contained in $\mathbf{D}_0$, which we denote by $\mathbf{B}$, and blocks that are not contained in $\mathbf{D}_0$, which we denote as $\mathbf{R}$. Thus, with appropriate permutations of the blocks of $\mathbf{D}'$ we can write $\mathbf{D}' = [\mathbf{B} \; \mathbf{R}]$. We perform the same permutation on $\mathbf{c}'$ resulting in vectors $\mathbf{b}$ and $\mathbf{r}$ such that $\mathbf{D}'\mathbf{c}' = \mathbf{B}\mathbf{b} + \mathbf{R}\mathbf{r}$. Since the vector $\mathbf{b}$ corresponds to blocks that are contained in $\mathbf{D}_0$, we can further decompose $\mathbf{b}$ as $\mathbf{b}^S + \mathbf{b}^{\overline{S}}$ where $\mathbf{b}^S$ indicates the values in $\mathbf{b}$ that correspond to columns in $\mathbf{D}_0^S$ supported on S, and $\mathbf{b}^{\overline{S}}$ correspond to the remaining columns. We similarly decompose the matrix $\mathbf{B}$ into $\mathbf{B}^S$ and $\mathbf{B}^{\overline{S}}$. Next we note that $\mathbf{Q}\mathbf{B}^S\mathbf{b}^S = \mathbf{Z}\mathbf{b}^S$, where $\mathbf{Z}$ is a $ks \times rs$ matrix consisting of blocks of size $s \times s$ that are either equal to the identity, or to zero. Here r is the number of blocks in $\mathbf{B}^S$, namely, the number of blocks shared by $\mathbf{D}_0$ and $\mathbf{D}'$. The blocks equal to the identity correspond to shared blocks. Substituting into (IV.32),

$$\mathbf{c}_0 = \mathbf{Q}(\mathbf{B}^{\overline{S}}\mathbf{b}^{\overline{S}} + \mathbf{R}\mathbf{r}) + \mathbf{Z}\mathbf{b}^S. \tag{IV.33}$$

Therefore, for the groups sparsity regularization term we have

$$\psi_{\mathcal{G}}(\mathbf{c}_0) \leq \psi_{\mathcal{G}}(\mathbf{b}^S) + \psi_{\mathcal{G}}(\mathbf{Q}\mathbf{B}^{\overline{S}}\mathbf{b}^{\overline{S}}) + \psi_{\mathcal{G}}(\mathbf{Q}\mathbf{R}\mathbf{r}). \tag{IV.34}$$

We now analyze the last two terms in (IV.34). To this end, we rely on the following lemma [25, Lemma 3].

Lemma 1. Let $\mathbf{v}$ be a vector with $\|\mathbf{v}_G\|_2 > 0$, for all g. Then for any matrix $\mathbf{Z}$ with appropriate dimensions, $\psi_{\mathcal{G}}(\mathbf{Z}\mathbf{v}) \leq \rho_c(\mathbf{Z})\psi_{\mathcal{G}}(\mathbf{v})$.

Since $\mathbf{B}^{\overline{S}}$ is contained in $\mathbf{D}_0^{\overline{S}}$ and $\mathbf{R}$ is contained in $\overline{\mathbf{D}}_0$, it follows from (IV.27) and (IV.28) that $\rho_c(\mathbf{Q}\mathbf{B}^{\overline{S}}) < 1$ and $\rho_c(\mathbf{Q}\mathbf{R}) < 1$. Combining this observation with Lemma 1 and (IV.34) we conclude that

$$\psi_{\mathcal{G}}(\mathbf{c}_0) < \psi_{\mathcal{G}}(\mathbf{b}^S) + \psi_{\mathcal{G}}(\mathbf{b}^{\overline{S}}) + \psi_{\mathcal{G}}(\mathbf{r}) = \psi_{\mathcal{G}}(\mathbf{c}'). \tag{IV.35}$$

The last equality is a result of the fact that $\mathbf{c}'$ is a concatenation of $\mathbf{b}^S, \mathbf{b}^{\overline{S}}$ and $\mathbf{r}$. Since $\psi_{\mathcal{G}}(\mathbf{c}_0) = \psi_{\mathcal{G}}(\mathbf{a}_0)$ and $\psi_{\mathcal{G}}(\mathbf{c}') = \psi_{\mathcal{G}}(\mathbf{a}')$ we have that $\psi_{\mathcal{G}}(\mathbf{a}_0) < \psi_{\mathcal{G}}(\mathbf{a}')$.

We now show in a similar fashion that $\|\mathbf{c}_0\|_1 < \|\mathbf{c}'\|_1$ or, equivalently, $\|\mathbf{a}_0\|_1 < \|\mathbf{a}'\|_1$. From [30, Theorem 3.3] we have that

$$\|\mathbf{c}_0\|_1 < \|\mathbf{Q}\mathbf{D}'\|_{1,1}\|\mathbf{c}'\|_1. \tag{IV.36}$$

This result is true as long as there is at least one column in $\mathbf{D}'$ that is not in $\mathbf{D}_0^S$. But this must be the case since by assumption, the columns of $\mathbf{D}_0^S$ are linearly independent. Therefore, if $\mathbf{D}'$ and $\mathbf{D}_0^S$ are equal, then we must have that $\mathbf{c}_0^S = \mathbf{c}'$. Combining (IV.36) with (IV.30) we conclude that $\|\mathbf{c}_0\|_1 < \|\mathbf{c}'\|_1$. Combining this result with (IV.35) we have,

$$\lambda\psi_{\mathcal{G}}(\mathbf{a}_0) + (1-\lambda)\left\|\mathbf{a}_0\right\|_1 \quad < \quad \lambda\psi_{\mathcal{G}}(\mathbf{a}') + (1-\lambda)\|\mathbf{a}'\|_1, \tag{IV.37}$$

so that $\mathbf{a}_0$ has the minimal objective from all possible solutions $\mathbf{a}'$ such that $\mathbf{x} = \mathbf{D}\mathbf{a}'$. ■

We conclude that we can guarantee recovery for every choice of λ as long as (IV.27)–(IV.29) are satisfied. We therefore turn to study these conditions in more detail.

Theorem 2. Let ${\mu_B}^{ss}, {\mu_B}^s, \nu^s$ be the sparse block-coherence measures defined in (IV.20),(IV.21) and (IV.22), and let ν be the sub-coherence of the dictionary $\mathbf{D}$ defined by (IV.16). Then the conditions in (IV.27)–(IV.29) are satisfied if

$$\begin{aligned}
\frac{kg{\mu_B}^s}{1-(s-1)\nu-(k-1)g{\mu_B}^{ss}} &\quad < \quad 1 && \text{(IV.38)}\\
\frac{kg\nu^s}{1-(s-1)\nu-(k-1)g{\mu_B}^{ss}} &\quad < \quad 1 && \text{(IV.39)}\\
\frac{ks\mu}{1-(s-1)\nu-(k-1)s\mu} &\quad < \quad 1. && \text{(IV.40)}
\end{aligned}$$

We also assume that all denominators are positive.

Recall from Proposition 1 that $g{\mu_B}^{ss} \leq s\mu$. Therefore, the denominators in (IV.38) and (IV.39) are smaller than that in (IV.40). However, there is no general ordering between the numerators. In the special case in which the individual dictionaries $\mathbf{D}_i$ consist of orthonormal columns, $\nu = \nu^s = 0$.

Proof: We begin by developing a bound on $\rho_c(\mathbf{Q}\overline{\mathbf{D}}_0)$. In [25] it is shown that $\rho_c(\cdot)$ is submultiplicative.[6] Therefore,

$$\rho_c(\mathbf{Q}\overline{\mathbf{D}}_0) \le \rho_c(((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1})\rho_c((\mathbf{D}_0^S)^T\overline{\mathbf{D}}_0). \tag{IV.41}$$

By definition,

$$\rho_c((\mathbf{D}_0^S)^T\overline{\mathbf{D}}_0) = \max_{j \notin \Lambda_0} \sum_{i \in \Lambda_0} \rho(\tilde{\mathbf{I}}^T\mathbf{D}_i^T\mathbf{D}_j), \tag{IV.42}$$

where Λ_0 is the set of indices ℓ for which $\mathbf{D}_\ell$ is in $\mathbf{D}_0$. Every element in the sum is bounded above by $g{\mu_B}^s$. Since Λ_0 contains k indices, we conclude that

$$\rho_c(\mathbf{Q}\overline{\mathbf{D}}) \le \rho_c(((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1})kg{\mu_B}^s. \tag{IV.43}$$

It remains to develop a bound for $\rho_c(((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1})$. To this end, we express $(\mathbf{D}_0^S)^T\mathbf{D}_0^S$ as $(\mathbf{D}_0^S)^T\mathbf{D}_0^S = \mathbf{I} + \mathbf{W}$, where $\mathbf{W}$ is a $(ks) \times (ks)$ matrix with blocks $\mathbf{W}_{\ell,r}$ of size $s \times s$ such that $\mathbf{W}_{\ell,r}[i,i] = 0$, for all i. This follows from the fact that the columns of $\mathbf{D}$ are normalized. Since $\mathbf{W}_{\ell,r} = [\mathbf{D}_0^S]_\ell^T[\mathbf{D}_0^S]_r$, for all $\ell \ne r$, and $\mathbf{W}_{r,r} = [\mathbf{D}_0^S]_r^T[\mathbf{D}_0^S]_r - \mathbf{I}_s$, we have

$$\begin{aligned}\rho_c(\mathbf{W}) &= \max_r \sum_\ell \rho(\mathbf{W}_{\ell,r}) \le \max_r \rho(\mathbf{W}_{r,r}) + \max_r \sum_{\ell \ne r} \rho(\mathbf{W}_{\ell,r}) &\text{(IV.44)}\\ &\le (s-1)\nu + (k-1)g{\mu_B}^{ss}. &\text{(IV.45)}\end{aligned}$$

By our assumptions, $(s-1)\nu + (k-1)g{\mu_B}^{ss} < 1$. Therefore, $\rho_c(\mathbf{W}) < 1$. We next use the following result from [25].

Lemma 2. Suppose that $\rho_c(\mathbf{W}) < 1$. Then $(\mathbf{I}+\mathbf{W})^{-1} = \sum_{k=0}^{\infty}(-\mathbf{W})^k$.

Using Lemma 2, we have that

$$\begin{aligned}\rho_c(((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1}) &= \rho_c\left(\sum_{k=0}^{\infty}(-\mathbf{W})^k\right) \overset{(a)}{\le} \sum_{k=0}^{\infty}(\rho_c(\mathbf{W}))^k\\ &= \frac{1}{1-\rho_c(\mathbf{W})} \overset{(b)}{\le} \frac{1}{1-(s-1)\nu-(k-1)g{\mu_B}^{ss}}. \end{aligned} \tag{IV.46}$$

Here, (a) is a consequence of $\rho_c(\mathbf{W})$ satisfying the triangle inequality and being submultiplicative, and (b) follows by using (IV.45). Combining (IV.46) with (IV.43), we obtain

$$\rho_c(\mathbf{Q}\overline{\mathbf{D}}_0) \le \frac{kg{\mu_B}^s}{1-(s-1)\nu-(k-1)g{\mu_B}^{ss}}, \tag{IV.47}$$

from which (IV.38) follows.

We now use the same technique to bound $\rho_c(\mathbf{Q}\mathbf{D}_0^{\overline{S}})$. Using the same method as above we will get a similar bound, with the only difference being in the term $\rho_c((\mathbf{D}_0^S)^T\mathbf{D}_0^{\overline{S}})$. By definition,

$$\rho_c((\mathbf{D}_0^S)^T\mathbf{D}_0^{\overline{S}}) = \max_{i \in \Lambda_0} \sum_{i \in \Lambda_0} \rho(\tilde{\mathbf{I}}^T\mathbf{D}_i^T\overline{\mathbf{D}_i}), \tag{IV.48}$$

where $\overline{\mathbf{D}_i}$ indicates the columns of $\mathbf{D}_i$ not chosen by $\tilde{\mathbf{I}}$. Each element $\rho(\tilde{\mathbf{I}}^T\mathbf{D}_i^T\mathbf{D}_i^{\overline{S}})$ is bounded by $g\nu^s$. Since there are k elements in the sum, $\rho_c((\mathbf{D}_0^S)^T\mathbf{D}_0^{\overline{S}}) \le kg\nu^s$ from which (IV.39) follows.

[6] A matrix norm is called submultiplicative if $\|\mathbf{Z}\mathbf{W}\| \le \|\mathbf{Z}\|\,\|\mathbf{W}\|$ for all matrixes $\mathbf{Z}, \mathbf{W} \in \mathbb{R}^{n \times n}$.

Finally, we use the same ideas to bound $\|\mathbf{Q}\overline{\mathbf{D}}\|_{1,1}$ and derive (IV.40). Specifically,

$$\|\mathbf{Q}\overline{\mathbf{D}}\|_{1,1} \leq \|((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1}\|_{1,1}\|(\mathbf{D}_0^S)^T\overline{\mathbf{D}}\|_{1,1}. \tag{IV.49}$$

Now

$$\|(\mathbf{D}_0^S)^T\overline{\mathbf{D}}\|_{1,1} = \max_{j\notin\Lambda_0}\sum_{i\in\Lambda_0}|\mathbf{d}_i^T\mathbf{d}_j|,$$

where Λ_0 is the set of active dictionary columns. Now, Λ_0 contains ks indices, so that $\|(\mathbf{D}_0^S)^T\overline{\mathbf{D}}\|_{1,1} \leq ks\mu$, which allows us to conclude that $\|\mathbf{Q}\overline{\mathbf{D}}\|_{1,1} \leq \|((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1}\|_{1,1}ks\mu$. It remains to develop a bound on $\|((\mathbf{D}_0^S)^T\mathbf{D}_0^S)^{-1}\|_{1,1}$. To this end we express $(\mathbf{D}_0^S)^T\mathbf{D}_0^S$ as $(\mathbf{D}_0^S)^T\mathbf{D}_0^S = \mathbf{I}+\mathbf{W}$, and bound $\|\mathbf{W}\|_{1,1}$, $\|\mathbf{W}\|_{1,1} \leq (s-1)\nu + (k-1)s\mu$. Continuing as before leads to (IV.40). ■

The above results are for the non-collaborative case. For the collaborative case there exist results that show that both the C-Lasso [9] and C-GLasso [24] will recover the true shared active set with a probability of error that vanishes exponentially with n. Since the in-group active sets are not necessarily equal for all samples in $\mathbf{X}$, C-HiLasso could only help in recovering the group sparsity pattern. Since the C-GLasso is a special case of C-HiLasso when $\lambda_1 = 0$, we can conjecture that when $\lambda_1 > 0$, the accuracy of the C-HiLasso in recovering the correct groups can only improve with larger n. Furthermore, since our results for the non-collaborative HiLasso improve on those of the non-collaborative C-GLasso, it is to be expected that the accuracy of C-HiLasso, for an appropriate $\lambda_1 > 0$, will be better than that of C-GLasso.

As an intuitive explanation of why this may happen, the proofs in [9] and [24] assume a continuous probability distribution on the non-zero coefficients of the signals, and give recovery results for the average case. On the other hand, the in-group sparsity assumption of C-HiLasso implies that only s out of g samples will be nonzero within each group. This implies that, for the same group sparsity pattern, there will be much less (exactly a fraction s/g) non-zero elements in the possible signals compared to the ones that can occur under the hypothesis of C-GLasso. Since any assumed distribution of the signals under the in-group sparsity hypothesis has to be concentrated on this much smaller set of possible signals, they should be easier to recover correctly from solutions to the C-HiLasso program, compared to the dense group case of C-GLasso.

V. Experimental results

In this section we show the strength of the proposed HiLasso and C-HiLasso models. We start by comparing our model with the standard Lasso and Group Lasso using synthetic data. We created $|\mathcal{G}|$ dictionaries, $\mathbf{D}_i$, with $g = 64$ atoms of dimension $m = 64$, with i.i.d. Gaussian entries. The columns were normalized to have unit ℓ_2 norm. Then we randomly chose two groups to be active at each time (on all the signals). Sets of $n = 200$ normalized testing signals were generated, one per active group, as linear combinations of $s \ll 64$ elements of the dictionaries, $\mathbf{x}_j^i = \mathbf{D}_i\mathbf{a}_j^i$. The mixtures were created by summing these signals and (eventually) adding Gaussian noise of standard deviation σ. The generated testing signals have a hierarchical sparsity structure and while they share groups, they do not necessarily share the sparsity pattern inside the groups.

We then built a single dictionary by concatenating the sub-dictionaries, $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_{|\mathcal{G}|}]$, and used it to solve the Lasso, group Lasso, HiLasso and C-HiLasso problems. Table I summarizes the Mean Squared Error (MSE) and Hamming distance of the recovered coefficient vectors. We observe that our model is able to exploit the hierarchical structure of the data as well as the collaborative structure. Group Lasso selects in general the correct blocks but it does not give a sparse solution within them. On the

$\sigma = 0.1$	41.7/22.0	117.3/361.6
	33.0/19.8	**16.3/13.3**
$\sigma = 0.2$	56.4/21.6	118.2/378.3
	39.9/22.7	**24.9** /**17.1**
$\sigma = 0.4$	96.5/22.7	137.8/340.3
	65.6/**19.5**	**59.5** /27.4

$s = 8$	38.8/22.0	118.4/318.2
	27.2/19.5	**9.6/16.2**
$s = 12$	120.0/36.2	116.6/350.4
	70.4/**26.5**	**41.3**/29.1
$s = 16$	164.1/43.9	109.3/338.6
	110.0/**32.2**	**55.1**/35.0

$\lvert\mathcal{G}\rvert = 4$	108.0/27.8	191.6/221.7
	100.9/**29.8**	**74.2** / 30.2
$\lvert\mathcal{G}\rvert = 8$	120.0/36.2	116.6/350.4
	70.4/**26.5**	**41.3** / 29.1
$\lvert\mathcal{G}\rvert = 12$	103.0/41.8	84.0/447.7
	66.2/**26.4**	**0.37**/ 29.8

TABLE I

Simulated signal results. In each table, each 2×2 cell contains, top to bottom, left to right, the result of Lasso, GLasso, HiLasso and C-HiLasso. For each method, the separation error (multiplied by 10^3) and Hamming distance are shown as (MSE/Hamming). In the first case (left) we vary the noise σ while keeping $|\mathcal{G}| = 8$ and $s = 8$ fixed. In the second and third cases we have $\sigma = 0$. For the second experiment (center) we fixed $|\mathcal{G}| = 8$ while changing s. In the third case we fix $s = 12$ and vary the number of groups $|\mathcal{G}|$. Bold indicates the best results, always obtained for the proposed models.

experiment	Lasso		Glasso		HiLasso		C-GLasso		C-HiLasso	
	APSNR	Hamm	APSNR	Hamm	APSNR	Hamm	APSNR	Hamm	APSNR	Hamm
1 digit	0.06	0.43	0.07	0.78	0.02	0.19	**0.01**	**0.02**	0.02	0.06
1 digit+n	0.08	1.31	0.08	0.87	0.04	0.48	0.05	0.25	**0.02**	**0.01**
2 digit	0.09	1.46	0.08	1.86	0.02	1.18	**0.01**	**0.74**	0.02	0.90
2 digit+n	0.11	2.21	0.08	1.99	0.04	1.46	0.09	1.60	**0.03**	**0.70**

TABLE II

Noisy digit mixtures results. Four different cases are shown: when each signal is a single digit and when it is the mixture of two different (randomly selected) digits, with and without additive Gaussian noise with standard deviation 10% of the peak value. For the 2 digits case, results are the average of 8 runs (in each round a new pair of digits was randomly selected). In the single digit case, the result is the average of the ten possible situations. Without noise, both C-GLasso and C-HiLasso yield very good results. However, in the noisy case, C-HiLasso is clearly superior, showing the advantage of adding regularization inside the groups from a robustness perspective. See also Figure 3.

other hand, Lasso gives a solution that has nonzero elements belonging to groups that were not active in the original signal, leading to a wrong model/class selection. HiLasso gives a sparse solution that picks atoms form the correct groups but still presents some minor mistakes. For the collaborative case, in all the tested configurations, no coefficients were selected outside the correct active groups, and the recovered coefficients are consistently the best ones.

In all the examples, and for each method, the regularization parameters were the ones for which the best results where obtained. One can scale the parameter λ_2 to account for different number of signals. This situation is analogous to a change in the size of the dictionary, thus, λ_2 should be proportional to the square root of the number of signals to code.

We then experimented with the USPS digits dataset, which has been shown to be well represented in the sparse modeling framework [34]. Here the signals are vectors containing the unwrapped gray intensities of 16×16 images ($m = 256$). We obtained each of the $n = 200$ samples in the testing data set as the mixture of two randomly chosen digits, one from each of the two drawn set of digits. In this case we only have ground truth at the group level. We measure the recovery performance in terms of the "separation error" [35], $\frac{1}{n|\mathcal{G}|}\sum_{i=1}^{|\mathcal{G}|}\sum_{j=1}^{n}\left\|\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i\right\|_2^2$, where $\mathbf{x}_j^i$ is the component corresponding to source i in the signal j, and $\hat{\mathbf{x}}_j^i$ is the recovered one.

Using the usual training-testing split for USPS we first learned a dictionary for each digit. We then created a single dictionary by concatenating them. In Table II we show the separation error obtained while summing two different numbers. We also consider the situation were only on digit is present. C-HiLasso

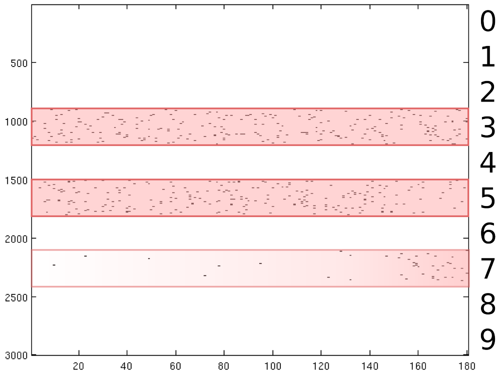


Fig. 3. In this example we used C-HiLasso to analyze mixtures where the data set contains different number and types of sources/classes. We used a set containing 180 mixture of digit images. The first 150 images are obtained as the sum/mixture of a number "3" and an number "5" (randomly selected). Each of the last 30 images in the set are the mixture of three numbers: "3" ,"5" and "7" (the 180 images are of course presented at random, the algorithm is not apriori aware which images contain 2 sources and which contain 3). The figure shows the active sets of the recovered coefficients matrix $\mathbf{A}$ as a binary matrix the same size as $\mathbf{A}$ (atom indexes in the vertical and sample indexes in the horizontal), where black dots indicate nonzero coefficients. C-HiLasso managed to identify the active blocks while the sub-dictionary corresponding to "7" is mostly active for the last 30 images. The accuracy of this result depends on the relationship between the sub-dictionaries corresponding to each digit.

automatically detects the number of sources while achieving the best recovery performance. As in the synthetic case, only the collaborative method was able to successfully detect the true active classes. In Figure 3 we relax the assumption that all the signals have to contain exactly the same type and amount of classes in the mixture, further demonstrating the flexibility of the proposed C-HiLasso model.

We also used the digits dataset to experiment with missing data. We randomly discarded an average of 60% of the pixels per mixed image and then applied C-Hilasso. The algorithm is capable of correctly detecting which digits are present in the images. Some example results for this case are shown in Figure 4. Note that this is a quite different problem than the one commonly addressed in the matrix completion literature. Here we do not aim to recover signals that all belong to a unique unknown sub-space, but signals that are the combination of two non-unique spaces to be automatically identified from the available dictionary. Such unknown spaces have common models/groups for all the signals in question (the coarse level of the hierarchy), but not necessarily the exact same atoms inside the groups and therefore not necessarily belong to the same sub-spaces. Both levels of the hierarchy are automatically detected, e.g., that the groups are those corresponding to "3" and "5," and the corresponding reconstructing atoms (sub-spaces) in each group, these last ones possibly different for each signal in the set. While we consider that the possible sub-spaces are to be selected from the provided dictionary (learned off-line from training data), in Section VI we discuss learning such dictionaries as part of the optimization as well (see also [36]). In such case, the standard matrix completion problem becomes a particular case of the C-HiLasso framework (with a single group and all the signals having the same active set, sub-space, in the group), naturally opening numerous theoretical questions for this new more general model.[7]

Finally, we compared the performance of C-HiLasso, Lasso, Group Lasso (GLasso) and C-GLasso (without hierarchy) in the task of separating mixed textures in an image. We chose 8 textures from the Brodatz dataset and trained one dictionary for each one of them (these form the 8 groups of the dictionary). Then we created an image as the sum of two textures (the testing images were not used in the training stage). One can think of this experiment as a generalization to the texture separation problem proposed

[7] Prof. Carin and collaborators have new results on the case of a single group and signals in possible different sub-spaces of the group, an intermediate model between standard matrix completion and C-HiLasso (personal communication).

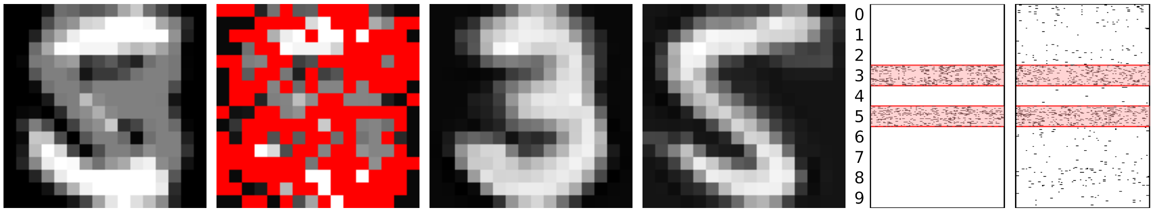


Fig. 4. Example of recovered digits (3 and 5) from a mixture with 60% of missing components. From left to right: noiseless mixture, observed mixture with missing pixels highlighted in red, recovered digits 3 and 5, and active set recovered for all samples using the C-HiLasso and Lasso respectively. In the last two figures, the active sets are represented as in Figure 3. The coefficients corresponding to the subdictionaries for digits 3 and 5 are marked as pink bands. Notice that the C-HiLasso exploits efficiently the hypothesis of collaborative group-sparsity, succeeding in recovering the correct active groups in all the samples. The Lasso, which lacks this prior knowledge, is clearly not capable of doing so, and active sets spread all over the groups.

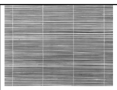
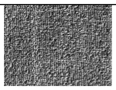
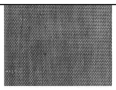
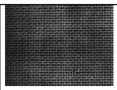
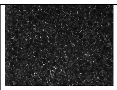
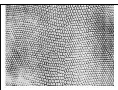
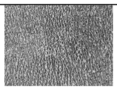
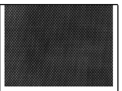
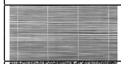
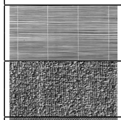
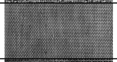
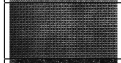
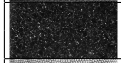
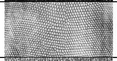
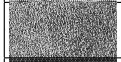
								
		19.6 16.7 19.3 **21.6**	27.4 21.3 21.6 **27.5**	22.0 21.1 19.0 **24.2**	27.2 23.3 23.3 **27.5**	20.7 17.6 18.8 **22.9**	19.7 13.5 19.9 **23.8**	31.5 23.7 25.7 **34.9**
	2.80 0.42 1.36 **0.00**		19.7 21.2 17.4 **21.7**	18.5 18.9 16.8 **19.9**	20.4 20.8 20.0 **21.1**	17.2 16.3 15.9 **18.5**	16.2 16.6 16.1 **17.5**	21.7 19.8 20.2 **27.3**
	0.33 0.25 2.06 **0.00**	3.65 **0.00** 2.67 0.02		22.8 **23.8** 18.0 23.7	24.6 22.1 20.8 **25.4**	19.8 19.5 16.7 **22.1**	17.9 18.5 17.0 **19.7**	26.8 20.3 19.9 **30.0**
	0.96 0.01 1.97 **0.00**	3.69 0.07 2.30 **0.00**	1.74 **0.00** 2.42 **0.00**		**23.1** 21.4 20.9 22.6	19.1 18.4 16.5 **20.1**	17.4 18.3 16.7 **19.7**	25.8 20.5 20.7 **30.0**
	1.02 1.00 2.25 **0.09**	3.55 1.00 2.52 **0.94**	1.42 1.00 3.39 **0.16**	2.25 1.00 2.85 **0.35**		20.7 21.2 19.2 **22.3**	19.2 20.6 19.7 **21.5**	28.3 22.0 23.9 **30.4**
	2.26 0.32 2.50 **0.00**	4.12 **0.53** 3.23 0.82	3.48 0.44 3.54 **0.20**	3.49 0.32 3.11 **0.01**	3.16 1.00 4.07 **0.40**		16.4 16.2 16.1 **17.9**	22.5 20.2 19.3 **25.7**
	4.37 1.39 2.51 **0.02**	4.47 **0.08** 2.39 0.22	4.09 0.13 2.42 **0.02**	4.23 0.12 2.76 **0.02**	4.20 1.00 2.24 **0.20**	4.42 0.42 2.96 **0.11**		20.0 19.5 19.9 **22.9**
	0.09 0.98 0.53 **0.00**	3.77 1.00 1.75 **0.01**	0.31 1.00 2.04 **0.00**	1.83 1.00 1.82 **0.00**	1.13 1.00 2.18 **0.00**	3.14 0.97 3.04 **0.24**	4.30 1.00 1.90 **0.18**	

TABLE III

TEXTURE SEPARATION RESULTS. THE ROWS AND COLUMNS INDICATE THE ACTIVE TEXTURES IN EACH CELL. THE UPPER TRIANGLE CONTAINS THE PSNR RESULTS, WHILE THE LOWER TRIANGLE SHOWS THE HAMMING ERROR IN THE GROUP-WISE ACTIVE SET RECOVERY. WITHIN EACH CELL, RESULTS ARE SHOWN FOR THE LASSO (TOP LEFT), GROUP LASSO (BOTTOM LEFT), COLLABORATIVE GROUP LASSO (TOP RIGHT) AND COLLABORATIVE HIERARCHICAL LASSO (BOTTOM RIGHT). THE BEST RESULTS ARE IN BLUE BOLD.

in [35] (without additive noise), where only two textures are present. The experiment was repeated for all possible combinations of two textures from the 8 possible ones, and the results are summarized in Table III. A detailed example is shown in Figure 5. For each algorithm, the best parameters were chosen using grid search, ensuring that those were not in the edges of the grid. For Lasso and C-HiLasso the best λ_1 is 0.0625. For GLasso and C-GLasso, the best λ_2 was, respectively, 0.05 and 75. From Table III we can conclude that the C-HiLasso is significantly better than the competing algorithms, both in PSNR of the recovered signals (we show the average PSNR of recovering both active signals), and in the average Hamming distance between the recovered group-wise active sets and the true ones. In the latter case we observe that, in many cases, the C-HiLasso active set recovery performance is perfect (Hamming distance 0) or near perfect, whereas the other methods seldom approach a Hamming distance lower than 1.

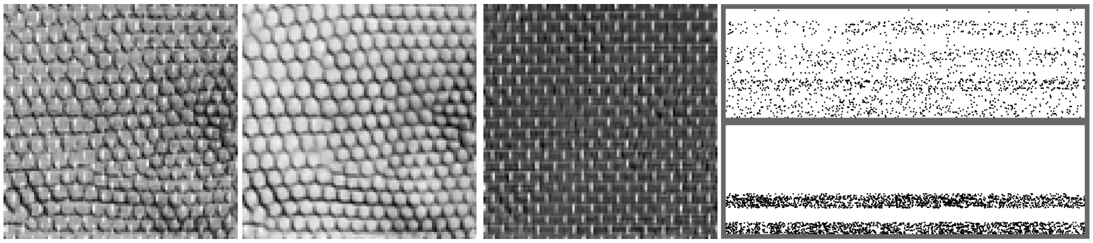


Fig. 5. Texture separation results. Left to right: sample mixture, corresponding C-HiLasso separated textures, and comparison of the active set diagrams obtained by the Lasso (as in Figure 4). The one for Lasso is shown on top, where all groups are wrongly active , and the one for C-HiLasso on bottom, showing that only the two correct groups are selected.

VI. Discussion

We introduced a new framework of collaborative hierarchical sparse coding, where multiple signals collaborate in their encoding, sharing code groups (models) and having (possible disjoint) sparse representations inside the corresponding groups. An efficient optimization approach was developed, which guarantees convergence to the global minimum, and examples illustrating the power of this framework were presented. At the practical level, we are currently working on the applications of this proposed framework in a number of directions, including collaborative instruments separation in music, signal classification, and speaker recognition, following the here demonstrated capability to collectively select the correct groups/models.

At the theoretical level, a whole family of new problems is opened by this proposed framework, some of which we already addressed in this work. A critical one is the overall capability of selecting the correct groups in the collaborative scenario, with missing information, and thereby of performing correct model selection and source identification and separation. Results in this direction will be reported in the future.

Finally, we have also developed an initial framework for learning the dictionary for collaborative hierarchical sparse coding, meaning the optimization is simultaneously on the dictionary and the code. As it is the case with standard dictionary learning, this is expected to lead to significant performance improvements (see [34] for the particular case of this with a single group active at a time).

Acknowledgments: Work partially supported by NSF, NSSEFF, ONR, NGA, and ARO. We thank Dr. Tristan Nguyen, when we presented him this model, he motivated us to think in a hierarchical fashion and to look at this as just the particular case of a fully hierarchical sparse coding framework. We thank Prof. Tom Luo and Gonzalo Mateos for invaluable help and advice on the use of `admom`. Finally, we thank Prof. Larry Carin, Dr. Guoshen Yu and Alexey Castrodad for very stimulating conversations, and for the fact that their own work (for LC, GY, and AC) also motivated in part the example with missing information.

References

[1] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *J. Royal Stat. Society, Series B*, vol. 68, pp. 49–67, 2006.

[2] R. Jenatton, J. Audibert, and F. Bach, “Structured variable selection with sparsity-inducing norms,” Tech. Rep. arXiv:0904.3523v1, INRIA, 2009.

[3] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Trans. Inform. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[4] J. Tropp, “Algorithms for simultaneous sparse approximation. part II: Convex relaxation,” *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[5] J. Tropp, A. Gilbert, and M. Strauss, “Algorithms for simultaneous sparse approximation. part I: Greedy pursuit,” *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[6] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” *IEEE Trans. Sig. Proc.*, vol. 53, no. 7, pp. 2477–2488, July 2005.

[7] J. Chen and X. Huo, “Theoretical results on sparse representations of multiple-measurement vectors,” *IEEE Trans. Sig. Proc.*, vol. 54, no. 12, pp. 4634–4643, Dec. 2006.

[8] M. Mishali and Y. C. Eldar, “Reduce and boost: Recovering arbitrary sets of jointly sparse vectors,” *IEEE Trans. Sig. Proc.*, vol. 56, no. 10, pp. 4692–4702, Oct. 2008.

[9] Y. C. Eldar and H. Rauhut, “Average case analysis of multichannel sparse recovery using convex relaxation,” to appear in *IEEE Trans. on Inform. Theory*.

[10] S. Wright, R. Nowak, and M. Figueiredo, “Sparse reconstruction by separable approximation,” *IEEE Trans. Sig. Proc.*, vol. 57, no. 7, pp. 2479–2493, 2009.

[11] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Comptutation: Numerical Methods*, Prentice Hall, 1989.

[12] T. Goldstein and S. Osher, “The split Bregman method for l1 regularized problems,” *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[13] J.-A. Bazerque, G. Mateos, and G. Giannakis, “Distributed lasso for in-network linear regression,” in *Proc. ICASSP*, Mar. 2010.

[14] P. Sprechmann, I. Ramirez, and G. Sapiro, “Collaborative hierarchical sparse modeling,” in *CISS*, Mar. 2010.

[15] J. Friedman, T. Hastie, and R. Tibshirani, “A note on the group lasso and a sparse group lasso,” preprint (2010), available at `http://www-stat.stanford.edu/~tibs`.

[16] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D. Noh, J. Pollack, and P. Wang, “Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer,” To appear in Annals of Applied Statistics.

[17] S. Kim and E. P. Xing, “Tree-guided group lasso for multi-task regression with structured sparsity,” in *ICML*, June 2010.

[18] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, “Proximal methods for sparse hierarchical dictionary learning,” in *ICML*, June 2010.

[19] J. Starck, M. Elad, and D. Donoho, “Image decomposition via the combination of sparse representations and a variational approach,” *IEEE Trans. Image Proc.*, vol. 14, pp. 1570–1582, 2004.

[20] R. Tibshirani, “Regression shrinkage and selection via the LASSO,” *J. Royal Stat. Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[21] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[22] D. Donoho, “Compressed sensing,” *IEEE Trans. on Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr 2006.

[23] B. Turlach, W. Venables, and S. Wright, “Simultaneous variable selection,” *Technometrics*, vol. 27, pp. 349–363, 2004.

[24] P. Boufounos, G. Kutyniok, and H. Rauhut, “Sparse recovery from combined fusion frame measurements,” arXiv:0912.4988v1.

[25] Y. C. Eldar, P. Kuppinger, H., and Bölcskei, “Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery,” to appear in *IEEE Trans. Sig. Proc.*

[26] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.,” *Comm. on Pure and Applied Mathematics*, vol. 57, pp. 1413–1457, 2004.

[27] R. Giryes, M. Elad, and Y. C. Eldar, “The projected GSURE for automatic parameter tuning in iterative shrinkage methods,” Submitted to *Applied and Computational Harmonic Analysis*.

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

[29] E. Candès, “The restricted isometry property and its implications for compressed sensing,” *C. R. Acad. Sci. Paris S’er. I Math.*, vol. 346, pp. 589–592, 2008.

[30] J. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[31] A. d’Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, “A direct formulation for sparse PCA using semidefinite programming,” *Neural Information Processing Systems*, vol. 17, 2004.

[32] H. Zou, T. Hastie, and R. Tibshirani, “Sparse principal component analysis,” *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, 2003.

[33] B. Moghaddam, Y. Weiss, and S. Avidan, “Spectral bounds for sparse PCA: Exact & greedy algorithms,” *Neural Information Processing Systems*, vol. 18, 2006.

[34] I. Ramirez, P. Sprechmann, and G. Sapiro, “Classification and clustering via dictionary learning with structured incoherence,” in *CVPR*, June 2010.

[35] N. Shoham and M. Elad, “Alternating KSVD-denoising for texture separation,” in *The IEEE 25-th Convention of Electrical and Electronics Engineers in Israel*, 2008.

[36] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and Lawrence Carin, “Non-parametric bayesian dictionary learning for analysis of noisy and incomplete images,” IMA Preprint, April 2010, `http://www.ima.umn.edu/preprints/apr2010/2307.pdf`.