

Combinatorial Selection and Least Absolute Shrinkage via the CLASH Algorithm

Anastasios Kyrillidis and Volkan Cevher

Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne

Abstract—The least absolute shrinkage and selection operator (LASSO) for linear regression exploits the geometric interplay of the ℓ_2 -data error objective and the ℓ_1 -norm constraint to arbitrarily select sparse models. Guiding this uninformed selection process with sparsity models has been precisely the center of attention over the last decade in order to improve learning performance. To this end, we alter the selection process of LASSO to explicitly leverage combinatorial sparsity models (CSMs) via the combinatorial selection and least absolute shrinkage (CLASH) operator. We provide concrete guidelines how to leverage combinatorial constraints within CLASH, and characterize CLASH’s guarantees as a function of the set restricted isometry constants of the sensing matrix. Finally, our experimental results show that CLASH can outperform both LASSO and model-based compressive sensing in sparse estimation.

I. INTRODUCTION

The *least absolute shrinkage and selection operator* (LASSO) is the *de facto* standard algorithm for regression [1]. LASSO estimates sparse linear models by minimizing the empirical data error via:

$$\hat{x}_{\text{LASSO}} = \arg \min \left\{ \|y - \Phi x\|_2^2 : \|x\|_1 \leq \lambda \right\}, \quad (1)$$

where $\|\cdot\|_r$ is the ℓ_r -norm. In (1), $\Phi \in \mathbb{R}^{m \times n}$ is the sensing matrix, $y \in \mathbb{R}^m$ are the responses (or observations), $x \in \mathbb{R}^n$ is the loading vector and $\lambda \in \mathbb{R}^{++}$ governs the sparsity of the solution. Along with many efficient algorithms for its solution, the LASSO formulation is now backed with a rather mature theory for the generalization of its solutions as well as its variable selection consistency [2]–[5].

While the long name attributed to (1) is apropos,¹ it does not capture the LASSO’s arbitrariness in subset selection via shrinkage to best explain the responses. In fact, this *uninformed* selection process not only prevents interpretability of results in many problems, but also fails to exploit key prior information that could radically improve learning performance. Based on this premise, approaches to guide the selection process of the LASSO are now aplenty.

Surprisingly, while the prior information in many regression problems generate fundamentally discrete constraints (e.g., on the sparsity patterns or the *support* of the LASSO solution),

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, and DARPA KeCoM program #11-DARPA-1055. VC also would like to acknowledge Rice University for his Faculty Fellowship.

¹Many of the optimization solutions to LASSO leverage *shrinkage* operations (e.g., as projections onto the ℓ_1 -ball) for sparse model *selections*. However, the geometric interplay of the ℓ_2 -data error objective and the ℓ_1 -norm constraint inherently promotes sparsity, independent of the algorithm.

the majority of the existing approaches that enforce such constraints in selection are inherently continuous. For instance, a prevalent approach is to tailor a *sparsity inducing* norm to the constraints on the support set (c.f., [6]). That is, we create a structured convex norm by *mixing* basic norms with weights over pre-defined groups or using the Lovász extension of non-decreasing submodular set functions of the support. As many basic norms have well-understood behavior in sparse selection, reverse engineering such norms is quite intuitive.

While such structure inducing, convex norm-based approaches on the LASSO are impressive, our contention in this paper is that, in order to truly make an impact in structured sparsity problems, we must fully leverage explicitly combinatorial approaches to guide LASSO’s subset selection process. To achieve this, we show how Euclidean projections with structured sparsity constraints correspond to an integer linear program (ILP), which can be exactly or approximately solved subject to matroid (via the greedy algorithm), and certain linear inequality constraints (via convex relaxation or multi-knapsack solvers). A key actor in this process is a polynomial-time combinatorial algorithm that goes beyond simple selection heuristics towards provable solution quality as well as runtime/space bounds.

Furthermore, we introduce our combinatorial selection and least absolute shrinkage (CLASH) operator and theoretically characterize its estimation guarantees. CLASH enhances the *model-based compressive sensing* (model-CS) framework [7] by combining ℓ_1 -norm and combinatorial constraints on the regression vector. Therefore, CLASH uses a combination of shrinkage and hard thresholding operations to significantly outperform the model-CS approach, LASSO, or continuous structured sparsity approaches in learning performance of sparse linear models. Furthermore, CLASH establishes a regression framework where the underlying tractability of approximation in combinatorial selection is directly reflected in the algorithm’s estimation and convergence guarantees.

The organization of the paper is as follows. In Sections II and III, we set up the notation and the exact projections with structured sparsity constraints. We develop CLASH in Section IV and highlight the key components of its convergence proof in Section V. We present numerical results in Section VI. We provide our conclusions in Section VII.

II. PRELIMINARIES

Notation: We use $[x]_j$ to denote the j -th element of x , and let x_i represent the i -th iterate of CLASH. The index set of n

dimensions is denoted as $\mathcal{N} = \{1, 2, \dots, n\}$. Given $\mathcal{S} \subseteq \mathcal{N}$, we define the complement set $\mathcal{S}^c = \mathcal{N} \setminus \mathcal{S}$. Moreover, given a set $\mathcal{S} \subseteq \mathcal{N}$ and a vector $x \in \mathbb{R}^n$, $(x)_{\mathcal{S}} \in \mathbb{R}^n$ denotes a vector with the following properties: $[(x)_{\mathcal{S}}]_{\mathcal{S}} = [x]_{\mathcal{S}}$ and $[(x)_{\mathcal{S}}]_{\mathcal{S}^c} = 0$. The support set of x is defined as $\text{supp}(x) = \{i : [x]_i \neq 0\}$. We use $|\mathcal{S}|$ to denote the cardinality of the set \mathcal{S} . The empirical data error is denoted as $f(x) \triangleq \|y - \Phi x\|_2^2$, with gradient defined as $\nabla f(x) \triangleq -2\Phi^T(y - \Phi x)$, where T is the transpose operation. The notation $\nabla_{\mathcal{S}} f(x)$ is shorthand for $(\nabla f(x))_{\mathcal{S}}$. \mathbb{I} represents the identity matrix.

Combinatorial notions of sparsity: We provide some definitions on combinatorial sparse models, and elaborate on a subset of interesting models with algorithmic implications.

Definition 1 (Combinatorial sparsity models (CSMs)). *We define a combinatorial sparsity model $\mathcal{C}_k = \{\mathcal{S}_m : \forall m, \mathcal{S}_m \subseteq \mathcal{N}, |\mathcal{S}_m| \leq k\}$ with the sparsity parameter k as a collection of distinct index subsets \mathcal{S}_m .*

Throughout the paper, we assume that any CSM \mathcal{C}_k is downward compatible, i.e., removing any subset of indices of any given element in \mathcal{C}_k , it is still in \mathcal{C}_k .

Properties of the regression matrix: Deriving approximation guarantees for CLASH behooves us to assume the restricted isometry property (RIP) (defined below) on the regression matrix Φ [8]. While the RIP and other similar conditions for deriving consistency properties of LASSO and its variants, such as the unique/exact representation property or the irrepresentable condition [5], [6], [9]–[11], are unverifiable *a priori* without exhaustive search, many *random* matrices satisfy them with high probability.

Definition 2 (RIP [7], [8]). *The regression matrix has the k -RIP with an isometry constant δ_k when*

$$(1 - \delta_k)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_k)\|x\|_2^2, \quad (2)$$

$\forall \text{supp}(x) \in \mathcal{C}_k$, where $\delta_k = \max_{\mathcal{S} \in \mathcal{C}_k} \|\Phi_{\mathcal{S}}^T \Phi_{\mathcal{S}} - \mathbb{I}\|_{2 \rightarrow 2}$, and $\Phi_{\mathcal{S}}$ is a submatrix of Φ as column-indexed by \mathcal{S} .

Here, we also comment on the scaling of (k, m, n) for the desired level of isometry. When the entries of Φ can be modeled as independent and identically distributed (iid) with respect to a sub-Gaussian distribution, we can show that $m = \mathcal{O}(\delta_k^{-2}(\log(2M) + k \log(12\delta_k^{-1})))$ with overwhelming probability [7]. Here, M is the minimum number of subspaces covering \mathcal{C}_k . While m explicitly depends on n , for certain restricted CSMs, such as the rooted connected tree of [7], this dependence can be quite weak, e.g., $m = \mathcal{O}(k)$.

III. EXACT AND APPROXIMATE PROJECTIONS ONTO CSMs

The workhorse of the model-CS approach is the following non-convex projection problem onto CSMs, as defined by \mathcal{C}_k , which is a basic subset selection problem:

$$\mathcal{P}_{\mathcal{C}_k}(x) = \arg \min_{w \in \mathbb{R}^n} \{\|w - x\|_2^2 : \text{supp}(w) \in \mathcal{C}_k\}, \quad (3)$$

where $\mathcal{P}_{\mathcal{C}_k}(x)$ is the projection operator. [7] shows that as long as $\mathcal{P}_{\mathcal{C}_k}(\cdot)$ is exactly computed in polynomial time for a CSM,

their sparse recovery algorithms inherit strong approximation guarantees for that CSM. To better identify the CSMs that live within the model-CS assumptions, we first state the following key observation—the proof can be found in [12].

Lemma 1 (Euclidean projections onto CSMs). *The support of the Euclidean projection onto \mathcal{C}_k in (3) can be obtained as a solution to the following discrete optimization problem:*

$$\text{supp}(\mathcal{P}_{\mathcal{C}_k}(x)) = \arg \max_{\mathcal{S} \in \mathcal{C}_k} F(\mathcal{S}; x), \quad (4)$$

where $F(\mathcal{S}; x) = \|x\|_2^2 - \|(x)_{\mathcal{S}} - x\|_2^2 = \sum_{i \in \mathcal{S}} |[x]_i|^2$ is the modular, variance reduction set function. Moreover, let $\hat{\mathcal{S}} \in \mathcal{C}_k$ be the minimizer of the discrete problem. Then, it holds that $\mathcal{P}_{\mathcal{C}_k}(x) = (x)_{\hat{\mathcal{S}}}$, which corresponds to hard thresholding.

The following proposition refines this observation to further accentuate the algorithmic implications for CSMs:

Proposition 1 (CSM projections via ILP's). *The problem (4) is equivalent to the following integer linear program (ILP):*

$$\text{supp} \arg \min_{\substack{z: [z]_i \in \{0,1\}, \\ \text{supp}(z) \in \mathcal{C}_k}} \{w^T z : [w]_i = -|[x]_i|^2\}, \quad (5)$$

where $[z]_i$, $(i = 1, \dots, n)$, are support indicator variables.

The proof of Proposition 1 is straightforward and is omitted.

Regardless of whether we use a dynamic program, a greedy combinatorial algorithm, or an ILP solver, the formulations (4) or (5) make the underlying tractability of the combinatorial selection explicit. We highlight this notion via the polynomial-time modular ϵ -approximation property (PMAP $_{\epsilon}$):

Definition 3 (PMAP $_{\epsilon}$ [12]). *A CSM has the PMAP $_{\epsilon}$ with constant ϵ , if the modular subset selection problem (4) or the ILP (5) admit an ϵ -approximation scheme with polynomial or pseudo-polynomial time complexity as a function of n , $\forall x \in \mathbb{R}^n$. Denoting the ϵ -approximate solution of (4) or (5) as $\hat{\mathcal{S}}_{\epsilon}$, this means $F(\hat{\mathcal{S}}_{\epsilon}; x) \geq (1 - \epsilon) \max_{\mathcal{S} \in \mathcal{C}_k} F(\mathcal{S}; x)$.*

In this paper, we focus and elaborate on CSMs with PMAP $_0$.

A. Example CSMs with PMAP $_0$

Matroids: By matroid, we mean that $\mathcal{C}_k = (\mathcal{N}, \mathcal{I})$ is a finite collection of subsets of \mathcal{N} that satisfies three conditions: (i) $\{\emptyset\} \in \mathcal{I}$, (ii) if \mathcal{S} is in \mathcal{I} , then any subset of \mathcal{S} is also in \mathcal{I} , and (iii) for $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{I}$ and $|\mathcal{S}_1| > |\mathcal{S}_2|$, there is an element $s \in \mathcal{S}_1 \setminus \mathcal{S}_2$ such that $\mathcal{S}_2 \cup \{s\}$ is in \mathcal{I} . As a simple example, the unstructured sparsity model (i.e., x is k -sparse) forms a *uniform matroid* as it is defined as the union of all subsets of \mathcal{N} with cardinality k or less. When \mathcal{C}_k forms a matroid, the greedy basis algorithm can efficiently compute (3) by solving (4) [13] where sorting and selecting the k largest elements in absolute value is sufficient to obtain the exact projection.

Moreover, it turns out that this particular perspective provides a principled and tractable approach to encode an interesting class of *matroid-structured sparsity models*. The recipe is quite simple: we seek the intersection of a structure provider

Algorithm 1: CLASH Algorithm

Input: $y, \Phi, \lambda, \mathcal{P}_{C_k}, \text{Tolerance } \eta, \text{MaxIterations}$
Initialize: $x_0 \leftarrow 0, \mathcal{X}_0 \leftarrow \{\emptyset\}, i \leftarrow 0$
repeat
1: $\mathcal{S}_i \leftarrow \text{supp}(\mathcal{P}_{C_k}(\nabla_{\mathcal{X}_i} f(x_i))) \cup \mathcal{X}_i$
2: $v_i \leftarrow \arg \min_{v: \|v\|_1 \leq \lambda, \text{supp}(v) \in \mathcal{S}_i} \|y - \Phi v\|_2^2$
3: $\gamma_i \leftarrow \mathcal{P}_{C_k}(v_i)$ with $\Gamma_i \leftarrow \text{supp}(\gamma_i)$
4: $x_{i+1} \leftarrow \arg \min_{x: \|x\|_1 \leq \lambda, \text{supp}(x) \in \Gamma_i} \|y - \Phi x\|_2^2$
5: $\mathcal{X}_{i+1} \leftarrow \text{supp}(x_{i+1})$
 $i \leftarrow i + 1.$
until $\|x_i - x_{i-1}\|_2 \leq \eta \|x_i\|_2$ or MaxIterations.

matroid (e.g., partition, cographic/graphic, disjoint path, or matching matroid) with the sparsity provider uniform matroid. While the intersection of two matroids is not a matroid in general, we can prove that the intersection of the uniform matroid with any other matroid satisfies the conditions above.

Linear support constraints: Many interesting CSMs C_k can be encoded using *linear support constraints* of the form:

$$C_k = \bigcup_{\mathfrak{Z} \in \mathfrak{Z}} \text{supp}(z), \quad \mathfrak{Z} := \{[z]_i \in \{0, 1\} : Az \leq b\},$$

where $[A, b]$ is an integral matrix, and the first row of A is all 1's and $[b]_1 = k$. As a basic example, the neuronal spike model of [14] is based on linear support constraints where each spike respects a minimum refractory distance to each other.

A key observation is that if each of the nonempty faces of \mathfrak{Z} contains an integral point (i.e., forming an integral polyhedra), then convex relaxation methods can *exactly* obtain the correct integer solutions in polynomial time. In general, checking the integrality of \mathfrak{Z} is NP-Hard. However, if \mathfrak{Z} is integral and non-empty for all integral b , then a necessary condition is that A be a totally unimodular (TU) matrix [13]. A matrix is totally unimodular if the determinant of each square submatrix is equal to 0, 1, or -1. Example TU matrices include interval, perfect, and network matrices [13]. As expected, the constraint matrix A of [14] is TU. Moreover, it is easy to verify that the sparse disjoint group model of [15] also defines a TU constraint, where groups have individual sparsity budgets.

B. How about PMAP $_\epsilon$?

For completeness and due to lack of space, we only mention PMAP $_\epsilon$, which extends the breath of the model-CS approach. For a detailed treatment of PMAP $_\epsilon$ and CLASH, c.f. [12], which describes multi-knapsack CSMs as a concrete example. Moreover, for many of the PMAP $_0$ examples above, we can employ ϵ -approximate—randomized—techniques to reduce computational cost.

IV. THE CLASH ALGORITHM

The new CLASH algorithm obtains approximate solutions to the LASSO problem in (1) with the added twist that the solution must live within the CSM, as defined by C_k :

$$\hat{x}_{\text{CLASH}} = \arg \min \{f(x) : \|x\|_1 \leq \lambda, \text{supp}(x) \in C_k\}. \quad (6)$$

When available, using the CSM constraint C_k in addition to the ℓ_1 -norm constraint enhances learning in two important ways. First, the combinatorial constraints restricts the LASSO solution to exhibit true model-based supports, *increasing the interpretability* of the solution without relaxing C_k into a convex norm. Second, it empirically requires much fewer number of samples to obtain the true solution than both the LASSO and the model-CS approaches.

We provide a pseudo-code of an example implementation of CLASH in Algorithm 1. One can think of alternative ways of implementing CLASH, such as single gradient updates in Step 2, or removing Step 4 altogether. While such changes may lead to different—possibly better—approximation guarantees for the solution of (6), we observe degradation in the empirical performance of the algorithm as compared to this implementation, whose guarantees are as follows:

Theorem 1 (Iteration invariant). *Let $x^* \in \mathbb{R}^n$ be the true vector that satisfies the constraints of (6) and let $y = \Phi x^* + \varepsilon$ be the set of observations with additive error $\varepsilon \in \mathbb{R}^m$. Then, the i -th iterate x_i of CLASH satisfies the following recursion:*

$$\|x_{i+1} - x^*\|_2 \leq \rho \|x_i - x^*\|_2 + c_1(\delta_{2k}, \delta_{3k}) \|\varepsilon\|_2$$

where $\rho \triangleq \frac{\delta_{3k} + \delta_{2k}}{\sqrt{1 - \delta_{2k}^2}} \sqrt{\frac{1 + 3\delta_{3k}^2}{1 - \delta_{3k}^2}}$ and $c_1(\delta_{2k}, \delta_{3k})$ is a constant defined in [16]. The iterations contract when $\delta_{3k} < 0.3658$.

Theorem 1 shows that the isometry requirements of CLASH are competitive with the mainstream hard thresholding methods, such as CoSaMP [17] and Subspace Pursuit [18], even though it incorporates the ℓ_1 -norm constraints, which, as Section VI illustrates, improves learning performance.

Remark 1. [Model mismatch and selection] *Let us assume a generative model $y = \Phi \beta + \tilde{\varepsilon}$. Let x^* be the best approximation of β in C_k within ℓ_1 -ball of radius λ . Then, we can show that the iteration invariant of Theorem 1 still holds with $\text{SNR} = \frac{\|x^*\|_2}{\|\varepsilon\|_2}$, where $\|\varepsilon\|_2 \leq \|\tilde{\varepsilon}\|_2 + \|\Phi(\beta - x^*)\|_2$, where the latter quantity (the impact of mismatch) can be analyzed using the restricted amplification property of Φ [7]. For instance, when C_k is the uniform sparsity model, then $\|\Phi(\beta - x^*)\|_2 \leq \sqrt{1 + \delta_k} \left(\|\beta - x^*\|_2 + \frac{\|\beta - x^*\|_1}{\sqrt{k}} \right)$, which should presumably be small if the model is selected correctly.*

In the absence of prior information, we automate the parameter selection by using the Donoho-Tanner phase transition [19] to choose the maximum k allowed for a given (m, n) -pair, and then by using cross validation to pick λ [20].

V. PROOF OF THEOREM 1

We sketch the proof of Theorem 1 a lá [17] and [21] assuming the general case of PMAP $_\epsilon$. The details of the proof can be found in an extended version of the paper [16].

Lemma 2 (Active set expansion - Step 1). *The support set \mathcal{S}_i , where $|\mathcal{S}_i| \leq 2k$, identifies a subspace in C_{2k} such that:*

$$\begin{aligned} \|(x_i - x^*)_{\mathcal{S}_i^c}\|_2 &\leq (\delta_{3k} + \delta_{2k} + \sqrt{\epsilon}(1 + \delta_{2k})) \|x_i - x^*\|_2 \\ &\quad + (\sqrt{2(1 + \delta_{3k})} + \sqrt{\epsilon(1 + \delta_{2k})}) \|\varepsilon\|_2 \end{aligned} \quad (7)$$

Lemma 2 states that, at each iteration, Step 1 of CLASH identifies a $2k$ support set such that the unrecovered energy of x^* is bounded. For $\epsilon = 0$, CLASH exactly identifies the support where the projected gradient onto \mathcal{C}_k can make most impact on the loading vector in the support complement of its current solution, which are subsequently merged together.

Lemma 3 (Greedy descent with least absolute shrinkage - Step 2). *Let \mathcal{S}_i be a $2k$ -sparse support set. Then, the least squares solution v_i in step 2 of Algorithm 1 satisfies*

$$\|v_i - x^*\|_2 \leq \frac{1}{\sqrt{1 - \delta_{3k}^2}} \|(x_i - x^*)_{\mathcal{S}_i^c}\|_2 + \frac{\sqrt{1 + \delta_{2k}}}{1 - \delta_{3k}} \|\epsilon\|_2.$$

We borrow the proof of Lemma 3 from [21]. This step improves the objective function $f(x)$ as much as possible on the active set in order to arbitrate the active set. The solution simultaneously satisfies the ℓ_1 -norm constraint.

Step 3 projects the solution onto \mathcal{C}_k , whose action is characterized by the following lemma. Here, we show the ϵ -approximate projection explicitly:

Lemma 4 (Combinatorial selection - Step 3). *Let v_i be a $2k$ -sparse proxy vector with indices in support set \mathcal{S}_i , \mathcal{C}_k be a CSM and γ_i the projection of v_i under \mathcal{C}_k . Then:*

$$\|\gamma_i - v_i\|_2^2 \leq (1 - \epsilon) \|(v_i - x^*)_{\mathcal{S}_i}\|_2^2 + \epsilon \|v_i\|_2^2.$$

Step 4 requires the following Corollary to Lemma 3:

Corollary 1 (De-bias - Step 4). *Let Γ_i be the support set of a proxy vector γ_i where $|\Gamma_i| \leq k$. Then, the least squares solution x_{i+1} in Step 4 satisfies*

$$\|x_{i+1} - x^*\|_2 \leq \frac{1}{\sqrt{1 - \delta_{2k}^2}} \|\gamma_i - x^*\|_2 + \frac{\sqrt{1 + \delta_k}}{1 - \delta_{2k}} \|\epsilon\|_2.$$

Step 4 de-biases the current result on the putative solution support. Its characterization connects Lemmas 3 and 4:

Lemma 5. *Let v_i be the least squares solution of the greedy descent step (step 5) and γ_i be a proxy vector to v_i after applying Combinatorial selection step. Then, $\|\gamma_i - x^*\|_2$ can be expressed in terms of the distance from v_i to x^* as follows:*

$$\begin{aligned} & \|\gamma_i - x^*\|_2 \\ & \leq \sqrt{1 + ((1 - \epsilon) + 2\sqrt{1 - \epsilon})\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon} \cdot \|v_i - x^*\|_2 \\ & + D_1 \|\epsilon\|_2 + D_2 \|x^*\|_2 + D_3 \sqrt{\|x^*\|_2 \|\epsilon\|_2}, \end{aligned} \quad (8)$$

where D_1, D_2, D_3 are constants depending on $\epsilon, \delta_{2k}, \delta_{3k}$.

Finally, the proof of Theorem 1 follows by concatenating Corollary 1 with Lemmas 2, 3, and 5, and setting $\epsilon = 0$.

VI. EXPERIMENTS

In the following experiments, we compare algorithms from the following list: (i) the LASSO algorithm [1], (ii) the Basis Pursuit DeNoising (BPDN) [22], (iii) the sparse-CLASH algorithm, where \mathcal{C}_k is the index set of k -sparse signals, (iv) the model-CLASH algorithm², which explicitly carries \mathcal{C}_k , and

(v) Subspace Pursuit (SP) algorithm [18], as integrated with the model-CS approach. We emphasize here that when $\lambda \rightarrow \infty$ in (6), CLASH must converge to the model-based SP solution.

The LASSO algorithm finds a solution to the problem defined in (1), where we use a Nesterov accelerated projected gradient algorithm. The BPDN algorithm in turn solves the following optimization problem:

$$\hat{x}_{\text{BPDN}} = \arg \min \{\|x\|_1 : \|\Phi x - y\|_2 \leq \sigma\}, \quad (9)$$

where σ represents prior knowledge on the energy of the additive noise term. To solve (9), we use the spectral projected gradient method SPGL1 algorithm [23].

In the experiments below, the nonzero coefficients of x^* are generated iid according to the standard normal distribution with $\|x^*\|_2 = 1$. The BPDN algorithm is given the true σ values. While CLASH is given the true value of k for the experiments below, additional experiments (not shown) shows that our phase transition heuristics is quite good and the mismatch is graceful as indicated in Remark 1. All the algorithms use a high precision stopping tolerance $\eta = 10^{-5}$.

Experiment 1: Improving simple sparse recovery. In this experiment, we generate random realizations of the model $y = \Phi x^* + \epsilon$ for $n = 800$. Here, Φ is a dense random matrix whose entries are iid Gaussian with zero mean and variance $1/m$. We consider two distinct generative model settings: (i) with additive Gaussian white noise with $\|\epsilon\|_2 = 0.05$, $m = 240$ and $k = 89$, and (ii) the noiseless model ($\|\epsilon\|_2 = 0$), $m = 250$ and sparsity parameter $k = 93$. For this experiment, we perform 500 Monte Carlo model realizations.

We sweep λ and illustrate the recovery performance of CLASH (6). Figures 1(a)-(b) illustrate that the combination of hard thresholding with norm constraints can *improve* the signal recovery performance significantly over convex-only and hard thresholding-only methods—both in noisy and noiseless problem settings. For $\|\epsilon\|_2 = 0$, CLASH perfectly recovers the signal when λ is close to the true value. When $\lambda \ll \|x^*\|_1$, the performance degrades due to the large norm mismatch.

Experiment 2: Improving structured sparse recovery We consider two signal CSMs: in the first model, we assume k -sparse signals that admit clustered sparsity with coefficients in C -contiguous blocks on an undirected, acyclic chain graph [24]. Without loss of generality, we use $C = 5$ (Figure 1(c)). The second model corresponds to a TU system where we partition the k -sparse signals into uniform blocks and force sparsity constraints on individual blocks; in this case, we solve the set optimization problem optimally via linear programming relaxation (Figure 1(d)). Here, the noise energy level satisfies $\|\epsilon\|_2 = 0.05$, and $n = 500$, $m = 125$, and $k = 50$. In both cases, we conduct 100 Monte Carlo iterations and perform sparse estimation for a range of λ values.

In Figure 1(c), we observe that clustered sparsity structure provides a distinct advantage in reconstruction compared to LASSO formulation and the sparse-CLASH algorithm. Furthermore, note that when λ is large, norm constraints have *no effect* and the model-CLASH provides essentially the same results as the model-CS approach [7]. On the other hand, the sparse-

²CLASH codes are available for MATLAB at <http://ions.epfl.ch/CLASH>.

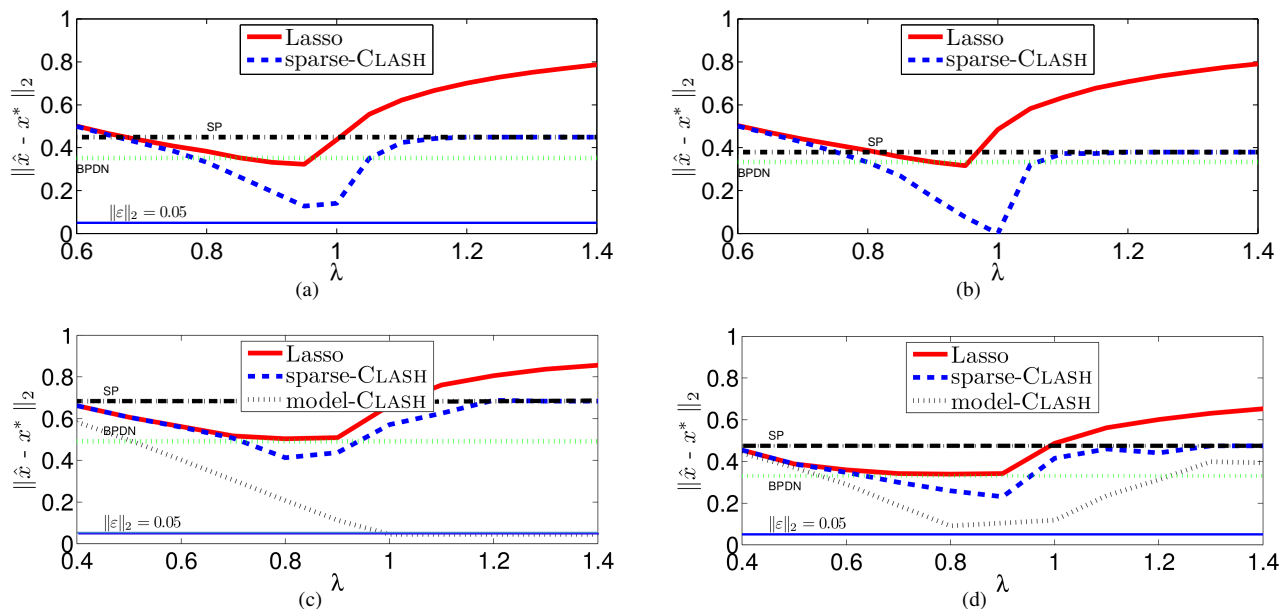


Fig. 1. Median values of signal error $\|\hat{x} - x^*\|_2$. Top row: simple sparsity model under noisy $\|\varepsilon\|_2 = 0.05$ (left column) and noiseless $\|\varepsilon\|_2 = 0$ (right column) settings. Bottom row: the (k, C) -clustered sparsity model (left column) and the TU model (right column).

CLASH improves significantly beyond the LASSO solution thanks to the ℓ_1 -norm constraint.

In Figure 1(d) however, the situation is radically changed: while the TU constraints enhance the reconstruction of model-CS approach over simple sparse recovery, the improvement becomes quite large as the ℓ_1 -norm constraint kicks in. We also observe the improvement in sparse-CLASH but it is not as accentuated as the model-CLASH.

VII. CONCLUSIONS

CLASH establishes a regression framework where efficient algorithms from combinatorial and convex optimization can interface for interpretable sparse solutions. Our experiments demonstrate that, while the model-based selection can greatly improve sparse recovery over the approaches based on uniform sparsity alone, the ℓ_1 -norm shrinkage operations have an undeniable, positive impact on the learning performance. Understanding the tradeoffs between the complexity of approximation and the recovery guarantees of CLASH is a promising theoretical and practical direction. Finally, joint combinatorial and norm constrained scenarios appear in many problems of interest such as portfolio optimization.

REFERENCES

- [1] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B*, 58(1):267–288, 1996.
- [2] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1 -ball for learning in high dimensions. In *ICML*, 2008.
- [3] P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.
- [4] M.J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (Lasso). *IEEE Trans. on Info. Theory*, 2009.
- [5] P. Zhao and B. Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.
- [6] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 2777–2824, 2011.
- [7] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. on Info. Theory*, 2010.
- [8] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Info. Theory*, 2006.
- [9] J.A. Tropp. Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. *Signal Processing*, 86(3):589–602, 2006.
- [10] L. Jacob, G. Obozinski, and J.P. Vert. Group lasso with overlap and graph lasso. In *ICML*, 2009.
- [11] F. Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, 2010.
- [12] A. Kyrillidis and V. Cevher. Sublinear time, approximate model-based sparse recovery for all. *EPFL Technical report*, 2011.
- [13] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.
- [14] C. Hegde, M.F. Duarte, and V. Cevher. Compressive sensing recovery of spike trains using a structured sparsity model. *SPARS*, 2009.
- [15] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv:1001.0736*, 2010.
- [16] A. Kyrillidis and V. Cevher. Combinatorial selection and least absolute shrinkage via the CLASH algorithm. *EPFL Technical Report*, 2011.
- [17] D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–312, 2008.
- [18] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. on Inf. Theory*, 2009.
- [19] D. L. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences*, 102, 2005.
- [20] R. Ward. Compressed sensing with cross validation. *IEEE Trans. on Info. Theory*, 2009.
- [21] S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Proceedings of the 13th International Conference on Approximation Theory*, 2010.
- [22] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.
- [23] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 2008.
- [24] V. Cevher, P. Indyk, C. Hegde, and R.G. Baraniuk. Recovery of clustered sparse signals from compressive measurements. *SAMPTA*, 2009.