BILEVEL FEATURE SELECTION IN NEARLY-LINEAR TIME

Chinmay Hegde

Electrical and Computer Engineering, Iowa State University

ABSTRACT

Selection of a small subset of informative features from data is a basic technique in signal processing, machine learning, and statistics. Joint selection of entire groups of features is desirable if the data features exhibit shared grouping structures. *Bilevel feature selection* constitutes a refinement of these ideas, producing a small subset of data features that themselves belong to a small number of feature groups. However, algorithms for bilevel feature selection suffer a computational cost that can be *cubic* in the size of the data, hence impeding their utility. In this paper, we propose an approach for bilevel feature selection that resolves this computational challenge. The core component of our approach is a novel fast algorithm for bilevel hard thresholding for a specific non-convex, discrete optimization problem. Our algorithm produces an approximate solution to this problem, but only incurs a nearly-linear running time. We extend this algorithm into a two-stage thresholding method that performs statistically as well as the best available methods for bilevel feature selection, but that also scales extremely well to massive dataset sizes.

1. INTRODUCTION

1.1. Motivation

Selection of a small subset of informative features from data is an important primitive in algorithms spanning several research areas. In signal processing, this technique forms the core of algorithms for *sparse recovery*, where the goal is to reconstruct a sparse signal from (possibly noisy) linear measurements. In statistics, this technique identifies a subset of relevant features to enhance prediction performance in multivariate linear regression and factor analysis.

However, data features often exhibit shared *grouping* structures, and informative features may co-occur as subsets of groups. For example, the nonzero coefficients of a sparse time-domain signal may occur as a single contiguous group. Here, it may be beneficial to jointly select entire (mutually exclusive) groups of correlated features. This approach is closely related to the *multitask learning problem* in machine learning applications.

Bilevel feature selection combines the best aspects of the above well-known techniques. The goal of bilevel feature selection is to identify a small subset of data features that *themselves* belong to a small number of feature groups. Similar to the sparse feature learning case, the bilevel approach encourages a parsimonious selection of data features; however, unlike the sparse case, it explicitly encourages intra-group feature selection. Due to its flexibility, bilevel feature selection has proved to be beneficial in applications spanning bioinformatics, web data mining, and medical imaging; see [1–8].

However, the benefits of bilevel feature selection are obtained at formidable computational costs. Algorithms for bilevel selection typically adopt one of two approaches. The first approach involves posing the feature selection as an optimization problem with a hybrid penalty that encourages sparsity as well as group-sparsity in the features. Representative examples in this class include the Sparse Group Lasso (SGL) [1], the SOGLasso [7], the group MCP approach [2], the group bridge approach [3], and the group exponential Lasso [9]. The second approach involves solving a combinatorial optimization problem using techniques from discrete programming. The Sparse Group Hard Thresholding (SGHT) approach of [6] is an example. The combinatorial approach is known to provide better statistical performance, as well as tighter (and intuitive) control over the sparsity as well as the number of groups in the selection. Unfortunately, the algorithmic cost of this optimization can be severe.

Concretely, consider a p-dimensional data vector x where the features are grouped into n (disjoint) groups of size g each (so that $p = ng$), and set parameters s_1 (desired sparsity) and s_2 (desired group sparsity). Then, SGHT returns an estimate $\widehat{x}$ such that $\|x - \widehat{x}\|_2$ is smallest among all vectors $\widehat{x}$ that are s_1-sparse and s_2-group sparse. However, the algorithm involves a challenging dynamic program (DP) that expends $O(ps_1s_2)$ running time as well as $O(ns_1s_2)$ memory. For massive data analysis problems where p, n, g are high and s_1, s_2 scale polynomially with p, the time and memory cost can be as high as *cubic* in the dataset size p. Therefore, this approach is not efficient for high dimensional problems.

1.2. Our contributions

In this paper, we develop a new approach for bilevel feature selection that resolves these computational challenges. Our approach is based on two algorithmic components:

1. As our primary contribution, we develop and analyze a new approximation algorithm for bilevel thresholding, that we call KEEPHEAVY, that runs in *nearly-linear time*. Concretely, for feature vectors of length p, our algorithm runs in time $O(p \log p)$ (independent of the sparsity and group sparsity parameters), and provably produces a solution with an objective value a (small) constant-factor of the optimum. Moreover, our algorithm requires only $O(p)$ memory and is simple to implement.
2. As our secondary contribution, we integrate this approximation method into an iterative two-stage hard thresholding algorithm for bilevel sparse linear regression. Executing this algorithm for a small number of iterations enables us to solve the overall problem of bilevel feature selection.

Table 1 summarizes the performance of our contributions relative to two representative previous approaches (SGL and SGHT). Numerical simulations reveal that our method achieves an identical statistical performance as the best available methods, but with at least an order of magnitude improvement in running time. For example, on a signal of size $p = 1,000,000$ with parameters $n = g = 1000$, we obtain at least a $30\times$ speedup with negligible loss in statistical efficiency. See Section 5 for details.

This work was supported in part by the National Science Foundation under grant CCF-1566281.

Table 1. *Comparison of different algorithms for bilevel selection. Data size is denoted by p, sparsity level by s_1, and group-sparsity level as s_2. $\widetilde{O}(\cdot)$ indicates scaling modulo polylogarithmic factors.*

Approach	Method	Reference	Runtime
SGL	Convex optimization	[1]	poly(p)
FISTA-SGHT	Dynamic program	[6]	$\widetilde{O}(ps_1s_2)$
Alg. 3	Approximation	This paper	$\widetilde{O}(p)$

1.3. Our techniques

Our two-stage hard thresholding algorithm is not particularly new; variations of this approach in the sparse recovery literature include CoSaMP [10], Subspace Pursuit [11], and iterative hard thresholding [12]. Our proposed algorithm can be viewed as an extension of Subspace Pursuit to the case of bilevel selection. This follows the approach of [13], which extended CoSaMP to the case of structured sparsity models; see, also, the recent works [14–17].

The key innovation is our new algorithm for bilevel thresholding. Instead of directly solving the discrete optimization posed in SGHT, we first perform a Lagrangian relaxation of the sparsity constraint to obtain a different (also combinatorial) problem. Somewhat surprisingly, we show the relaxed problem can be solved in linear $O(p)$ time and only $O(p)$ extra space. We solve a carefully constructed sequence of such Lagrangian relaxations ($\log(p)$ in number). Combined with a particular termination step, we obtain a final estimate $\widehat{x}$ which satisfies a constant approximation-factor guarantee[1]. Therefore, the overall thresholding algorithm runs in $O(p \log p)$. This constitutes a polynomial improvement over the existing FISTA-SGHT approach in the regime when s_1, s_2 are $\Theta(p)$.

Due to space constraints, we merely include our algorithms, theorems, and representative experiments; please see [18] for details.

2. BACKGROUND

We will concern ourselves with feature vectors $x \in \mathbb{R}^p$. For a given set of indices S, $x_S \in \mathbb{R}^p$ denotes the vector x restricted to the indices in S. Throughout the paper, we assume that the coordinates of any given vector $x \in \mathbb{R}^p$ can be partitioned into n disjoint (non-overlapping) groups, $G_1, G_2, \ldots, G_n$, of equal cardinality g, so that $p = ng$. Formally, $G_i \subseteq [p]$ denotes the indices belonging to group i for $i = 1, \ldots, n$; $|G_i| = g$; and $G_i \cap G_j = \emptyset$.

The symbol $\|\cdot\|_2$ denotes the ℓ_2-norm. The symbol $I(\cdot)$ denotes the indicator function that equals 1 if its argument is true, and 0 otherwise. The support of x, denoted by $\mathrm{supp}(x) \subseteq [p]$ is the set of indices corresponding to the nonzero coefficients of x. We call x as being s_1-sparse if at most s_1 of its p coefficients are nonzero, i.e., $\sum_{i=1}^{p} \mathbb{I}(|x_i| > 0) \leq s_1$. We call x as being s_2-group-sparse if at most s_2 of its n feature groups contain at least one nonzero coefficient, i.e., $\sum_{j=1}^{n} \mathbb{I}(\left\|x_{G_j}\right\|_2 > 0) \leq s_2$.

Following [6], we define the (s_1, s_2)-thresholding problem as follows. Given an input vector $x \in \mathbb{R}^p$, the goal is to find:

$$\widehat{x} = \arg\min_{v} \ \|x - v\|_2^2 \qquad (1)$$
$$\text{s. t. } \sum_{i=1}^{p} \mathbb{I}(|v_i| > 0) \leq s_1, \sum_{j=1}^{n} \mathbb{I}(\left\|v_{G_j}\right\|_2 > 0) \leq s_2.$$

[1]Due to the relaxation of the sparsity parameter, our estimate $\widehat{x}$ contains αs_1 nonzeros, where α is a small number bigger than 1.

The (s_1, s_2)-thresholding problem is a challenging combinatorial optimization problem owing to the presence of the indicator constraints. Interestingly, the challenge is due to the simultaneous presence of sparsity and grouping constraints; removing either one of the constraints results in a far easier problem that can be solved via standard thresholding. To the best of our knowledge, the only (exact) algorithm for solving (1) is the SGHT approach of [6].

Our focus in this paper will be to use bilevel feature selection to solve more general *linear regression* problems. In particular, given a linear operator $A \in \mathbb{R}^{m \times p}$ and a response vector $y \in \mathbb{R}^m$, the goal now is to find an $x \in \mathbb{R}^p$ that is s_1-sparse and s_2-group-sparse that minimizes the regression error in the least-squares sense:

$$\widehat{x} = \arg\min_{x} \ \|y - Ax\|_2^2, \qquad (2)$$
$$\text{s. t. } \sum_{i=1}^{p} \mathbb{I}(|x_i| > 0) \leq s_1, \sum_{j=1}^{n} \mathbb{I}(\left\|x_{G_j}\right\|_2 > 0) \leq s_2.$$

In the high-dimensional regime, the length of the response m is much smaller than the data size p. Enforcing both constraints is known to considerably improve regression performance in such regimes. Algorithms for solving problems of the form (2) have been long studied in the literature. One approach is to replace the nonconvex combinatorial constraints in (2) by suitable convex surrogate penalties. The Sparse Group Lasso [1] is one of the earliest such convex formulations. Subsequently, researchers have developed new surrogate penalties (many of them nonconvex) with associated estimation algorithms. Instances of these newer approaches including the group MCP [2, 4] and the group bridge [3].

An alternative approach is to use the SGHT algorithm of [6] with FISTA iterations. This approach directly attempts to solve the nonconvex problem using a projected gradient descent-like method. The number of iterations required for the algorithm to converge depends on several parameters such as the initialization, choice of step-size, and acceleration. Nevertheless, each iteration involves a bilevel thresholding problem of the form (1), and the cost per iteration can be as high as cubic in the size of the data. Interestingly, the SGHT algorithm provides the best statistical performance, as well as comparable running time, when contrasted against the best penalty-based approaches.

3. APPROXIMATE BILEVEL PROJECTION

In this section, we propose a fast, simple, and approximate algorithm for solving the (s_1, s_2)-projection problem. Observe that the optimization problem (1) is a *hard thresholding problem*; the goal is to discover a signal $\widehat{x}$ with support Ω, containing at most s_1 elements grouped into at most s_2 groups, such that $\widehat{x}_\Omega = x_\Omega$ and $\widehat{x}_{\Omega^c} = 0$.

We introduce some more notation to simplify the discussion. Let $\mathrm{card}(\cdot)$ denote the cardinality of a set. Let $\mathbb{G}(\cdot)$ denote the number of "active" groups for any given support Ω:

$$\mathbb{G}(\Omega) = \sum_{j=1}^{n} \mathbb{I}(\Omega \cap G_j \neq \emptyset).$$

If OPT is the value of the objective function at the optimum, then we can write (1) as:

$$OPT = \min_{\Omega \subseteq [p]} \ \|x - x_\Omega\|_2^2 \qquad (3)$$
$$\text{s. t. } \mathrm{card}(\Omega) \leq s_1, \ \mathbb{G}(\Omega) \leq s_2.$$

Algorithm 1 Solving the Lagrangian relaxation

function KEEPHEAVY(x, s, λ)
  for $i \in [p]$ **do**
    $b_i \leftarrow x_i^2 - \lambda$
  for $j \in [n]$ **do**
    $c_j \leftarrow \sum_{i \in G_j} \max(0, b_i)$
  $S \leftarrow$ Top-s indices of c_j
  $\Omega = \bigcup_{j \in S} \text{supp}(\max(0, b_{G_j}))$
  return Ω

Algorithm 2 Approximate bilevel projection

function BILEVELAPPROX($x, s_1, s_2, \gamma, \varepsilon$)
  $\lambda_l \leftarrow \|x\|_\infty^2, \quad \lambda_r \leftarrow 0$
  while $\lambda_l - \lambda_r > \varepsilon / s_1$ **do**
    $\lambda_m \leftarrow \frac{\lambda_l + \lambda_r}{2}$
    $\widehat{\Omega} \leftarrow$ KEEPHEAVY(x, s_2, λ_m)
    if $\text{card}(\widehat{\Omega}) \geq s_1$ and $\text{card}(\widehat{\Omega}) \leq (1+\gamma)s_1$ **then**
      return $\widehat{\Omega}$
    else if $\text{card}(\widehat{\Omega}) < s_1$ **then**
      $\lambda_l \leftarrow \lambda_m$
    else
      $\lambda_r \leftarrow \lambda_m$
  return $\widehat{\Omega} \leftarrow$ KEEPHEAVY(x, s_2, λ_l)

Instead of solving this problem using the dynamic program (DP) prescribed in [6], we will adopt an alternate approach. We first perform a *Lagrangian relaxation* of the s_1-sparsity constraint on Ω, while retaining the S_2 group sparsity constraint. That is, for a fixed parameter $\lambda > 0$, consider the modified problem:

$$OPT_{rel} = \min_{\Omega \subseteq [p]} \|x - x_\Omega\|_2^2 + \lambda \,\text{card}(\Omega), \qquad (4)$$
$$\text{s.t. } \mathbb{G}(\Omega) \leq s_2.$$

The choice of parameter λ controls the trade-off between the quality of approximation, versus the sparsity, of the estimated signal x_Ω.

First, we develop an algorithm to solve (4), explained in pseudocode form as Alg. 1 (KEEPHEAVY). We also obtain the following theorem regarding the correctness and runtime of KEEPHEAVY.

Theorem 1. *Let $x \in \mathbb{R}^p$. Then, for any fixed s and λ,* KEEPHEAVY *runs in $O(p)$ time and returns a support $\widehat{\Omega}$, satisfying $\mathbb{G}(\widehat{\Omega}) \leq s$, such that*

$$OPT_{rel} = \left\|x - x_{\widehat{\Omega}}\right\|_2^2 + \lambda \,\text{card}(\widehat{\Omega}).$$

Therefore, using KEEPHEAVY, we can solve (4) in linear time. Since the Lagrangian relaxation gives us only indirect control over the sparsity of the solution via the parameter λ, we need to choose λ carefully. In order to achieve this, we adopt a binary search procedure. If the sparsity of the output support $\widehat{\Omega}$ is too high relative to s_1, we increase λ, while if it is too low, we decrease λ. Algorithm 2 describes this procedure in pseudocode.[2] In addition to x, s_1 and s_2, and p, the algorithm requires two additional parameters γ and ε; γ denotes the aforementioned blow-up factor in the sparsity, while ε is a precision parameter dictating the number of iterations required for the binary search.

In fact, we can show that not only does such a binary search produces a support $\widehat{\Omega}$ of acceptable sparsity, but it also ensures that the error of approximation is close to the optimum of the original (constrained) problem (1). We prove the following:

[2] A very similar technique based on binary search has appeared before in [15], but for a rather different discrete optimization problem.

Algorithm 3 Two-stage Bilevel Hard Thresholding

function TSBHT($y, A, s_1, s_2, \gamma, \varepsilon, T$)
  $x_0 \leftarrow 0$
  for $t \leftarrow 1, \ldots, T$ **do**
    $v \leftarrow A^T(y - Ax_{t-1})$
    $\Gamma \leftarrow \text{supp}(\widehat{x}_{t-1})$
    $\Gamma \leftarrow \Gamma \cup$ BILEVELAPPROX($v, 2s_1, 2s_2, \gamma, \varepsilon$)
    $w_\Gamma \leftarrow A_\Gamma^\dagger y, \quad w_{\Gamma^C} \leftarrow 0$
    $\Omega \leftarrow$ BILEVELAPPROX($w, s_1, s_2, \gamma, \varepsilon$)
    $z_\Omega \leftarrow A_\Omega^\dagger y, \quad z_{\Omega^C} \leftarrow 0$
    $\widehat{x}_t \leftarrow z$
  return $\widehat{x} \leftarrow x_T$

Theorem 2. *Let $x \in R^p$ be an input vector, and s_1, s_2 be integer-valued parameters. Let γ, $\varepsilon > 0$ be real-valued parameters. Then,* BILEVELAPPROX *returns a support $\widehat{\Omega}$ such that* $\text{card}(\widehat{\Omega}) \leq (1+\gamma)s_1$ *and $\mathbb{G}(\widehat{\Omega}) \leq s_2$. Moreover, $\widehat{\Omega}$ satisfies:*

$$\left\|x - x_{\widehat{\Omega}}\right\|_2^2 < \left(1 + \frac{1}{\gamma}\right) OPT + \varepsilon \,.$$

The algorithm runs in time $O(p \log \frac{\|x\|_\infty s_1}{\varepsilon})$. Assuming that x is provided with $O(\log p)$ bits of precision, this simplifies to $O(p \log p)$.

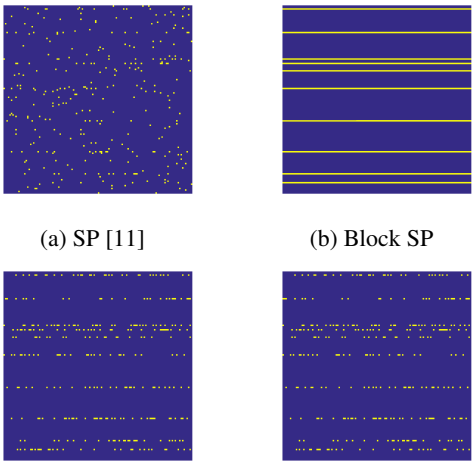
Theorem 2 implies that Algorithm 2 is *bi-criteria* approximate. It provides an answer which is within a small constant factor of the optimum and also incurs a small blowup in the sparsity of the returned solution. In our experiments, we have observed that these bounds are somewhat loose, and in practice we get better approximation ratios. The upshot, of course, is that the overall algorithm runs in (nearly) linear time and requires only $O(p)$ extra space.

4. BILEVEL FEATURE SELECTION

Armed with our fast (approximate) algorithm for bilevel hard thresholding, we are now ready to solve general feature selection problems of the form (2). The standard approach is to integrate the approximate bilevel projection step into a gradient descent-type algorithm, similar to the approach of [6]. Here, we propose an alternate approach that we call *Two-stage bilevel hard thresholding*.

Two-stage thresholding approaches have been developed in several different contexts [10–12]. A prototypical two-stage thresholding algorithm is Subspace Pursuit [11], developed for solving sparse linear regression problems. Our algorithm can be viewed as an extension of Subspace Pursuit to the case. Similar extensions of a (slightly) different two-stage thresholding method have been known to provide very good numerical performance [13, 15, 17, 19].

We describe the algorithm in pseudocode form as Alg. 3. The algorithm takes in as input a response vector $y \in \mathbb{R}^m$ and a design matrix $A \in R^{m \times p}$. It also requires sparsity and group-sparsity parameters s_1, s_2. The algorithm returns an estimate $\widehat{x}$ that has at most $(1+\gamma)s_1$ nonzeros within s_2 groups. In both theory and practice, Alg. 3 exhibits *linear* convergence, and therefore the overall running time is proportional (up to logarithmic factors) to the cost of matrix-vector multiplication using the matrix A. For special matrices that support fast multiplication (e.g., subsampled Fourier matrices), this cost is nearly-linear in p. We defer a rigorous analysis of Alg. 3 to the full version of this paper [18].



(a) SP [11] (b) Block SP

(c) FISTA-SGHT [6] (d) Alg. 3

Fig. 1. *Example support recovery results of a signal of size* $p = 128 \times 128$ *using* $m = 1140$ *measurements.*

Algorithm	SP [11]	Block SP	SGHT [6]	Alg. 3
MSE (dB)	2.1	9.6	51.64	**52.49**
Runtime (sec)	0.0035	0.0008	5.0101	**0.0510**

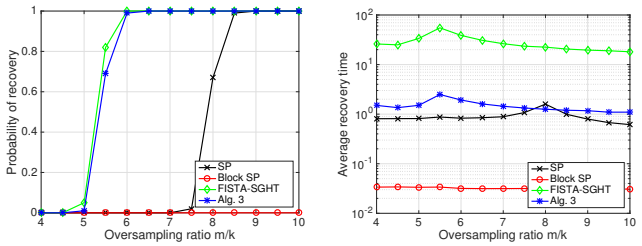
Table 2. *Quantitative comparisons of algorithms for data in Fig. 1.*

5. NUMERICAL RESULTS

In this section, we demonstrate the efficacy of our algorithms via a series of numerical experiments. All experiments were conducted on an iMac desktop computer equipped with a 3.5GHz Intel Core i5 processor and 16GB RAM. In each of our experiments below, we set the relaxation parameter γ to 1.1, the iteration parameter T to 40, and the binary search parameter ε to the machine precision in MATLAB (2.2×10^{-16}). For demonstration purposes, we also include the performance of previously proposed algorithms for feature selection. Due to space (and time) constraints, we limit our comparisons to only a few representative methods, and defer a more thorough set of numerical comparisons as future work. Among these representative algorithms are: (a) Subspace Pursuit (SP) [11], an algorithm for sparse recovery that ignores the s_2-grouping constraint. (b) A block version of Subspace Pursuit, that performs group-sparse recovery but ignores the s_1-sparsity constraints. (c) FISTA-SGHT, the dynamic programming-based method of [6] that enforces both constraints.

We first demonstrate the statistical efficiency of our proposed algorithms. We generate a synthetic test feature vector $x \in \mathbb{R}^p$ of length $p = 16,384$, comprising $n = 128$ groups of size $g = 128$ each. The signal x contains $s_1 = 300$ nonzeros where all the nonzeros belong to $s_2 = 10$ groups. The nonzero coefficients of x are independently drawn from a standard normal distribution. We construct a *design* matrix $A \in \mathbb{R}^{m \times p}$ whose entries are drawn from a normal distribution of variance $1/m$. Then, we compute linear observations $y = Ax$. The goal is to recover x from knowledge of y and A, with as few observations as possible.

Figure 1 displays the estimated supports of the reconstructed signals from $m = 1140$ measurements. At this level of sampling, we see that Subspace Pursuit fails to reliably recover the signal support. Block Subspace Pursuit recovers the groups correctly, but fails to distill out the actual features of the signal. In contrast, both FISTA with SGHT and Alg. 3 are able to accurately recover the support.



(a) Recovery (b) Runtime

Fig. 2. *Results of a Montecarlo experiment illustrating the performance of Alg. 3. (a) Comparison of probability of recovery. (b) Comparison of running time.*

Table 2 reports some quantitative comparisons. We measure (in decibels), the normalized estimation error (NMSE), measured as $-20 \log 10 \left(\frac{\|x - \widehat{x}\|_2}{\|x\|_2} \right)$, as well as the (cumulative) running time of the projection step in the different algorithms. We observe that our proposed algorithm yields the best error performance (and comparable to SGHT). The key advantage is in the running time; our algorithm achieves a $98\times$ speedup over SGHT.

We now measure the *empirical sample complexity* of the different algorithms. For any given recovery algorithm, the empirical sample complexity represents the minimal number of observatinos needed to reliably recover the signal.For this experiment, we set $p = 1,000,000$ where $n = g = 1000$, and leave s_1 and s_2 the same as before. Similar to the above example, we generate a test signal x of length p and parameters s_1 and s_2. Here, A is constructed by randomly subsampling m rows of a $p \times p$ discrete Fourier transform (DFT) matrix.

Figure 2 plots the results of a Monte Carlo experiment measuring the performance of the different recovery algorithms — Subspace Pursuit, Block Subspace pursuit, FISTA-SGHT, and Alg. 3 — for increasing values of m. Each data point in the plot was calculated by averaging over 100 independent trials. For this plot, "success" was declared if the ℓ_2-error of the recovery estimate $\widehat{x}$ was within 5% of the ℓ_2-norm of x. The intent is to measure, for each algorithm, the transition point for the ratio m/s that demarcates the distinction between recovery success and failure.

Figure 2(a) shows that Alg. 3 matches the success rate of FISTA with SGHT, *despite* the fact that it only performs approximate bilevel thresholding in each of its iterations. By leveraging both across- and within-group sparsity in the feature vector x, both SGHT and Alg. 3 far outperform the methods that use only one of the two two constraints. In this particular setting, "conventional" sparse recovery requires an empirical sample complexity that is about 50% worse than either SGL or our method. From these plots, we also observe that the block version of Subspace Pursuit yields poor recovery results. This is not surprising; this algorithm retains *all* features in the selected groups, increasing the sparsity of the target by a factor g.

It has been shown in [6] that FISTA with SGHT yields superior statistical performance compared to several previous state-of-the-art methods for bilevel regression (including [1, 3, 20]). By extension, our method can be expected to perform competitively relative to these methods. Figure 2(b) compares the projection running time (in seconds) of the different algorithms. We see that our proposed method achieves at least a $30\times$ speedup over FISTA-SGHT. An open question is whether ideas from our approach can be used to speed up other methods such as [1–4].

6. REFERENCES

[1] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group LASSO," *J. Comput. and Graphical Stat.*, vol. 22, no. 2, pp. 231–245, 2013.

[2] S. Ma and J. Huang, "Penalized feature selection and classification in bioinformatics," *Briefings in bioinformatics*, vol. 9, no. 5, pp. 392–403, 2008.

[3] J. Huang, S. Ma, H. Xie, and C.-H. Zhang, "A group bridge approach for variable selection," *Biometrika*, vol. 96, no. 2, pp. 339–355, 2009.

[4] P. Breheny and J. Huang, "Penalized methods for bi-level variable selection," *Statistics and its interface*, vol. 2, no. 3, pp. 369, 2009.

[5] J. Huang, P. Breheny, and S. Ma, "A selective review of group selection in high-dimensional models," *Stat. Sci.*, vol. 27, no. 4, 2012.

[6] S. Xiang, T. Yang, and J. Ye, "Simultaneous feature and feature group selection through hard thresholding," in *Proc. KDD*, 2014, pp. 532–541.

[7] N. Rao, C. Cox, R. Nowak, and T. Rogers, "Sparse overlapping sets lasso for multitask learning and its application to FMRI analysis," in *Proc. Adv. Neur. Inf. Proc. Sys.*, 2013, pp. 2202–2210.

[8] N. Rao, R. Nowak, C. Cox, and T. Rogers, "Classification with sparse overlapping groups," *arXiv preprint arXiv:1402.4512*, 2014.

[9] P. Breheny, "The group exponential LASSO for bi-level variable selection," *Biometrics*, vol. 71, no. 3, pp. 731–740, 2015.

[10] Deanna Needell and Joel Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, 2009.

[11] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[12] P. Jain, A. Tewari, and P. Kar, "On iterative hard thresholding methods for high-dimensional m-estimation," in *Proc. Adv. Neur. Inf. Proc. Sys.*, 2014, pp. 685–693.

[13] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[14] C. Hegde, P. Indyk, and L. Schmidt, "Approximation-tolerant model-based compressive sensing," in *Proc. ACM Symp. Discrete Alg. (SODA)*, Jan. 2014.

[15] C. Hegde, P. Indyk, and L. Schmidt, "A fast approximation algorithm for tree-sparse recovery," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, June 2014.

[16] C. Hegde, P. Indyk, and L. Schmidt, "Nearly linear-time model-based compressive sensing," in *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, July 2014.

[17] C. Hegde, P. Indyk, and L. Schmidt, "Approximation algorithms for model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 5129–5147, 2015.

[18] C. Hegde, "Bilevel feature selection in nearly-linear time," Tech. Rep., Iowa State University, 2016.

[19] C. Hegde, P. Indyk, and L. Schmidt, "A nearly linear-time framework for graph-structured sparsity," in *Proc. Int. Conf. Machine Learning*, July 2015.

[20] L. Yuan, Y. Wang, P. Thompson, V. Narayan, and J Ye, "Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data," *NeuroImage*, vol. 61, no. 3, pp. 622–632, 2012.