# Efficient Sparse Modeling With Automatic Feature Grouping

Leon Wenliang Zhong and James T. Kwok

***Abstract*— For high-dimensional data, it is often desirable to group similar features together during the learning process. This can reduce the estimation variance and improve the stability of feature selection, leading to better generalization. Moreover, it can also help in understanding and interpreting data. Octagonal shrinkage and clustering algorithm for regression (OSCAR) is a recent sparse-modeling approach that uses a $\ell_1$-regularizer and a pairwise $\ell_\infty$-regularizer on the feature coefficients to encourage such feature grouping. However, computationally, its optimization procedure is very expensive. In this paper, we propose an efficient solver based on the accelerated gradient method. We show that its key proximal step can be solved by a highly efficient simple iterative group merging algorithm. Given $d$ input features, this reduces the empirical time complexity from $O(d^2 \sim d^5)$ for the existing solvers to just $O(d)$. Experimental results on a number of toy and real-world datasets demonstrate that OSCAR is a competitive sparse-modeling approach, but with the added ability of automatic feature grouping.**

***Index Terms*— Accelerated gradient descent, feature grouping, sparse modeling, structured sparsity.**

## I. Introduction

REAL-WORLD data are often high dimensional and contain spurious features. Sparse modeling, which selects a relevant subset of features while learning the model, is thus becoming indispensable. It has been widely used in diverse application areas such as computer vision [1], image analysis [2], [3], signal processing [4], and bioinformatics [5].

Lasso [6] is the most popular sparse-modeling algorithm. It uses the squared error and an $\ell_1$-regularizer. As is well known, this regularizer induces sparsity in the solution. However, in the presence of highly correlated features, it tends to arbitrarily select only one of them [7]. Consequently, estimation can be unstable, and the resultant model difficult to interpret.

Another deficiency with lasso is that it cannot find feature groups. In general, feature grouping can reduce the estimator's variance [8] and improve the stability of feature selection [9]. It also helps to gain additional insight on the underlying data generation process, which is useful, for example, in the finding of coregulated genes [10] and understanding of protein–protein interaction networks [8]. Note that this is different from the group lasso and its variants [11], which require the feature groups be known in advance and may not be feasible in many applications. Moreover, group lasso cannot encourage coefficients in the same group to have similar magnitudes.

A simple approach to find feature groups is by first performing clustering on the data, and then input the cluster centers as new features to a supervised learner [12], [13]. However, this clustering step cannot take supervised information into account. A more desirable approach is to perform feature clustering and classifier training simultaneously. Dettling and Bühlmann [10] used a heuristic strategy that grows and prunes the feature groups incrementally. Jörnsten and Yu [9] combined feature clustering and classification into a single minimum description length code, and then use a search procedure to find the best model.

The explicit search for feature groups is a combinatorial optimization problem. Alternatively, one can encourage group formation by using an appropriate regularizer in the optimization problem. A well-known approach is the elastic net [7], which uses a combination of $\ell_1$- and $\ell_2$-regularizers. Let $\beta_i$s be the feature coefficients of the regression model. The elastic net encourages $\beta_i$ to be close to $\beta_j$ for highly correlated features $i$, $j$. Empirically, this is better than lasso in grouping features, but is still less effective than the above grouping methods [13]. In situations where the features are ordered in some meaningful way, the fused lasso [14] directly encourages the successive feature coefficients to be similar by using the regularizer $\sum_{i=2}^{d} |\beta_i - \beta_{i-1}|$. However, oftentimes such an ordering does not exist naturally. One then needs to estimate it by hierarchical clustering or multidimensional scaling, which incurs additional overhead and may also introduce error.

Recently, interest has focused on the more challenging problem where features are not ordered [8], [15]–[17]. Similar to the fused lasso, they all try to pull two feature coefficients $\beta_i, \beta_j$ together. For example, Wu *et al.* [16] proposed an $\ell_1 + \ell_\infty$ regularizer. However, the $\ell_\infty$-norm only penalizes those features with maximum absolute value. Shen and Huang proposed the grouping pursuit [8], which uses the regularizer $\sum_{j<j'} G(\beta_j - \beta_{j'})$ where $G(z) = \lambda$ (a regularization parameter) if $|z| > \lambda$; and $|z|$ otherwise. While the fused lasso only requires the successive coefficients to be similar, grouping pursuit requires all $(\beta_j, \beta_{j'})$ pairs to be similar. Computationally, as $G(\cdot)$ is nonconvex, this leads to a sequence of expensive difference of convex programs. Moreover, it cannot obtain sparse solutions. The clustered lasso [17] uses the regularizer

Manuscript received October 5, 2011; revised May 10, 2012; accepted May 10, 2012. Date of publication July 12, 2012; date of current version August 1, 2012. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region under Grant 614311.

The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: wzhong@cse.ust.hk; jamesk@cse.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2200262

$\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{i<j} |\beta_i - \beta_j|$, and can be viewed as a sparse convex variant of [8]. However, given $d$ features, its solver handles the $O(d^2)$ pairwise penalty terms explicitly, and thus is not scalable for large $d$. Moreover, its convergence analysis assumes that the sample covariance is nonsingular, which may not be feasible especially when the dataset is high dimensional.

In this paper, we will focus on the octagonal shrinkage and clustering algorithm for regression (OSCAR) method [15]. With a novel pairwise $\ell_\infty$ norm on the feature coefficients, it encourages both sparsity and equality of coefficients for highly correlated features. Feature groups are automatically discovered simultaneously with regression shrinkage, without the need to prespecify the grouping structure as in group lasso. Moreover, since the coefficients for the grouped features are tied, the resultant model has smaller model complexity and is less prone to overfitting.

Despite these advantages, the optimization problem of OSCAR, though still convex, is much more challenging. Bondell and Reich [15] proposed two solvers which unfortunately are not scalable for large $d$, and their experiments are limited to small feature sets. Alternatively, as OSCAR's pairwise $\ell_\infty$ norm is simply the sum of $\ell_\infty$ norms over groups of two variables, one can use the network flow algorithm recently proposed in [18]. However, this algorithm is designed for general overlapping groups but not tailored for OSCAR. Because of the $O(d^2)$ number of groups in OSCAR, each iteration will involve solving a maxflow problem on a canonical graph $G = (V, E)$ with $|V| = |E| = O(d^2)$, and results in a complexity of $O(d^5)$ [19].

In this paper, we propose an accelerated gradient algorithm [20] that is tailored for OSCAR's optimization problem. By using a simple group merging algorithm, the key proximal step can be solved exactly and efficiently in $O(d\log(d))$ time. Empirically, it is even faster and scales only as $O(d)$. Hence, the proposed algorithm is particularly efficient on high-dimensional datasets. Compared with the other sparse models, the proposed method achieves comparable accuracy with state-of-the-art methods on both regression and classification problems, but with the added ability of automatic feature grouping.

The rest of this paper is organized as follows. In Section II, we first give a brief review on OSCAR and accelerated gradient methods. Section III then describes the proposed solver. Experimental results are presented in Section IV, and Section V gives some concluding remarks. A preliminary conference version of this paper has been reported in [21]. All the proofs are in the Appendix.

## II. Related Work

### A. OSCAR [15]

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the input data matrix (with each row being an instance) and $\mathbf{y} \in \mathbb{R}^n$ be the corresponding output. We assume that $\mathbf{y}$ is centered (i.e., $\sum_{i=1}^n y_i = 0$) and each column of $\mathbf{X}$ is standardized (i.e., for column $j$, $\sum_{i=1}^n x_{ij} = 0$ and $\text{var}(x_{ij}) = 1$). OSCAR is formulated as the following optimization problem:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{i<j} \max\{|\beta_i|, |\beta_j|\} \quad (1)$$

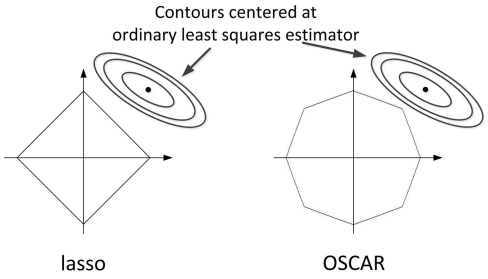



Fig. 1. Constraint regions for lasso and OSCAR.

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_d]^T$, and $\lambda_1, \lambda_2$ are regularization parameters. The OSCAR regularizer thus consists of two parts: an $\ell_1$-regularizer which encourages sparsity as in lasso and a pairwise $\ell_\infty$-regularizer which encourages every coefficient pair $|\beta_i|, |\beta_j|$ to be equal (Fig. 1). This can encourage highly correlated features to be grouped together [15]. On sorting the indices such that $|\beta_1| \geq |\beta_2| \geq \cdots \geq |\beta_d|$, the regularizer can also be rewritten as $\sum_{i=1}^d [\lambda_1 + (d-i)\lambda_2]|\beta_i|$.

In [15], (1) is considered in the equivalent form $\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$: $(1-c)\|\boldsymbol{\beta}\|_1 + c\sum_{i<j} \max\{|\beta_i|, |\beta_j|\} \leq t$, for the appropriate values of $c$ and $t$. Two solvers are proposed. The first one splits each $\beta_j$ into positive and negative parts, as $\beta_j = \beta_j^+ - \beta_j^-$ ($\beta_j^+ \geq 0, \beta_j^- \geq 0$), and then introduces $d(d-1)/2$ variables $\eta_{jk}$ ($1 \leq j < k \leq d$) for the pairwise maxima. This leads to a huge quadratic programming (QP) with $(d^2+3d)/2$ variables and $d^2+d+1$ linear constraints. The second solver uses a sequential QP (SQP) algorithm in which constraints are gradually added. It starts with a QP problem with $2d+1$ constraints. At each iteration, a new constraint is added and the new QP is solved. This is iterated until a solution is found. However, the total number of constraints can be $O(d!)$ in the worse case. Hence, both solvers cannot be used on high-dimensional data.

### B. Accelerated Gradient Methods

Gradient methods are well known for their simplicity and scalability. However, a major drawback is that they have slow convergence, especially when the objective function is nonsmooth. In the past decades, attempts have been made to accelerate gradient methods. Nesterov pioneered the "optimal method" for smooth optimization, which achieves the optimal convergence rate for a black-box model [22]. Subsequent works [20] extended this for composite optimization

$$\min_{\boldsymbol{\beta}} \; f(\boldsymbol{\beta}) + r(\boldsymbol{\beta}) \quad (2)$$

where $f(\boldsymbol{\beta})$ is convex with Lipschitz continuous gradient and $r(\boldsymbol{\beta})$ is convex but nonsmooth. Recently, accelerated gradient methods have been successfully used in many learning problems, where $f(\boldsymbol{\beta})$ is usually taken as the loss and $r(\boldsymbol{\beta})$ as the regularizer. Examples include lasso [20] and its variants [18], [23], trace norm minimization [24], and other sparsity-inducing norms [25].

While gradient methods perform descent by simply using the (sub)gradient of the nonsmooth objective in (2), accelerated methods solve the following optimization problem

**Algorithm 1** FISTA Algorithm for OSCAR

**Initialize:** $\hat{\boldsymbol{\beta}}^1 \leftarrow \boldsymbol{\beta}^0 \in \mathbb{R}^n$, $\tau_1 \leftarrow 1, t \leftarrow 1$.
**repeat**
  $\boldsymbol{\beta}^t \leftarrow \arg\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}^t)$. {proximal step (which is solved by using Algorithm 2)}
  $\tau_{t+1} \leftarrow \frac{1+\sqrt{1+4\tau_t^2}}{2}$.
  $\hat{\boldsymbol{\beta}}^{t+1} \leftarrow \boldsymbol{\beta}^t + \left(\frac{\tau_t - 1}{\tau_{t+1}}\right)(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1})$.
  $t \leftarrow t + 1$.
**until** convergence (the stopping criterion can be based on the duality gap computed in Algorithm 3).
Output $\boldsymbol{\beta}^t$.

(often called the *proximal* step):

$$\arg\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}^t) \equiv f(\hat{\boldsymbol{\beta}}^t) + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^t)^T \nabla f(\hat{\boldsymbol{\beta}}^t) + \frac{L}{2}\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^t\|^2 + r(\boldsymbol{\beta}) \tag{3}$$

where $\hat{\boldsymbol{\beta}}^t$ is the current estimate at iteration $t$, and $L$ is the Lipschitz constant[1] of $\nabla f(\boldsymbol{\beta})$. Note that $Q(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}^t)$ is a linear approximation of the smooth component $f(\boldsymbol{\beta})$ while leaving the nonsmooth component $r(\boldsymbol{\beta})$ intact. Problem (3) is also a common construct in the proximal methods [25] and many recent stochastic (sub)gradient methods [26].

The subsequent update step depends on the specific accelerated algorithm. In this paper, we will adopt the fast iterative shrinkage-thresholding algorithm (FISTA) [20], which is shown in Algorithm 1. While traditional gradient methods have a slow convergence rate of $O(1/\sqrt{N})$, where $N$ is the number of iterations, FISTA converges as $O(1/N^2)$. However, this requires that the crucial proximal step in (3) to be solved efficiently and exactly.

### C. OSCAR and Structured Sparsity-Inducing Norm

The OSCAR regularizer is an example of structured sparsity-inducing norm [25], [27]. Recently, Mairal *et al.* [18] considered structured norms of the general form

$$\sum_{g \in \mathcal{G}} \eta_g \|\boldsymbol{\beta}_g\|_\infty \tag{4}$$

where $\mathcal{G}$ is an arbitrary set of overlapping groups of indices in $\{1, 2, \ldots, d\}$, $\boldsymbol{\beta}_g$ is the subvector of $\boldsymbol{\beta}$ indexed by group $g$, and $\eta_g$ is the corresponding weight. They developed an efficient algorithm (called ProxFlow) which is based on accelerated gradient methods. As discussed in Section II-B, its success thus relies on the efficient computation of the proximal step. It is shown in [18] that the dual of this step can be reformulated as a quadratic min-cost flow algorithm on a canonical graph $G(V, E)$, where $|V| = |\mathcal{G}| + d$ and $|E| = |\mathcal{G}| + \sum_{g \in \mathcal{G}} |g| + d$. The worst case complexity for solving this maxflow problem is $O(|V|^2|E|^{(1/2)})$ [19], though empirically it can be much faster.

[1] In other words, $\|\nabla f(\boldsymbol{\beta}) - \nabla f(\tilde{\boldsymbol{\beta}})\| \leq L\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$ for every $\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}$.

Obviously, (4) admits the OSCAR regularizer in (1) as a special case, and thus ProxFlow can be readily used for its optimization. However, because of the $O(d^2)$ number of groups in OSCAR's pairwise $\ell_\infty$-regularizer, the canonical graph in ProxFlow's maxflow problem has $|V| = |E| = O(d^2)$. Consequently, each proximal step takes $O(|V|^2|E|^{(1/2)}) = O(d^5)$ time. As will be shown in Section IV, empirically this is only as fast as the QP and SQP solvers in [15].

Recently, Bach [28] proposed an interesting connection between structured sparsity-inducing norms and submodular functions. Specifically, the OSCAR regularizer $r(\boldsymbol{\beta})$ can be viewed as the Lovász extension of the submodular function $R(A) = \lambda_2(d|A| - \sum_{i=1}^{|A|} i) + \lambda_1|A|$, where $A \subseteq \{1, 2, \ldots, d\}$ is a subset of indices. Thus, the proximal step can be cast as submodular function minimization, in which standard solvers can be used. However, these solvers either have high complexity [e.g., $O(d^6)$] or no complexity bound at all (e.g., the minimum-norm-point algorithm). Moreover, as these solvers are generic, they can be rather inefficient for a specific model such as OSCAR.

## III. Efficient Proximal Step for OSCAR

The OSCAR objective in (1) can be decomposed into the form of (2), with

$$f(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{5}$$

which is smooth, and

$$r(\boldsymbol{\beta}) = \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{i<j} \max\{|\beta_i|, |\beta_j|\} \tag{6}$$

which is nonsmooth. The following proposition shows that the proximal step (3) can be written in a more compact form. All the proofs are given in the Appendix.

*Proposition 1:* With $f(\cdot)$ in (5) and $r(\cdot)$ in (6), the proximal step (3) can be rewritten as

$$\min_{\mathbf{z} \geq \mathbf{0}} F(\mathbf{z}) \equiv (\mathbf{z} - \mathbf{a})^T\mathbf{z} + \hat{r}(\mathbf{z}) \tag{7}$$

where $\mathbf{z} = [z_1, z_2, \ldots, z_d]^T$ and $\mathbf{a} = [a_1, a_2, \ldots, a_d]^T$, with

$$z_i = |\beta_i| \tag{8}$$

$$a_i = \frac{2}{L}\left|L\hat{\beta}_i^t - [\nabla f(\hat{\boldsymbol{\beta}}^t)]_i\right| \tag{9}$$

and

$$\hat{r}(\mathbf{z}) = \frac{2}{L}\left(\lambda_1 \sum_{i=1}^{d} z_i + \lambda_2 \sum_{i<j} \max\{z_i, z_j\}\right). \tag{10}$$

Without loss of generality, we assume that the indices $i \in \{1, 2, \ldots, d\}$ have been permuted such that

$$a_1 \geq a_2 \geq \cdots \geq a_d \geq 0. \tag{11}$$

Since $(\mathbf{z} - \mathbf{a})^T\mathbf{z}$ is strongly convex, we immediately obtain the following corollary.

*Corollary 1:* Problem (7) has a unique optimal solution.

### A. Properties of the Optimal Solution of (7)

Since the OSCAR regularizer encourages $|\beta_i| = |\beta_j|$ (or, equivalently, $z_i = z_j$), we expect some of $z_1, \ldots, z_d$ will be lumped into groups. Let $\mathbf{z}^*$ be the optimal solution of (7). In this section, we present the three properties of this optimal $\mathbf{z}^*$, namely:

1) $z_i^*$s are in nonincreasing order (Proposition 2);
2) the value in each group of $\mathbf{z}^*$ can be easily determined (Proposition 3);
3) the groups in $\mathbf{z}^*$ are "coherent" (Proposition 4).

*1) $z_i^*$s are Nonincreasing:* With the $a_i$s sorted in decreasing order, the following proposition shows that $z_i^*$s will also be arranged in the same order.

*Proposition 2:* At the optimal solution

$$z_1^* \geq z_2^* \geq \cdots \geq z_d^* \geq 0. \tag{12}$$

Moreover, if $a_i = a_j$, then $z_i^* = z_j^*$.

As $z_i^*$ is the $i$th largest value in $\{z_1^*, z_2^*, \ldots, z_d^*\}$, it can be seen that its contribution to (10) is $(2/L)(\lambda_1+\lambda_2(d-i))z_i^*$. We can then replace $\mathbf{z} \geq \mathbf{0}$ in (7) by the more explicit constraint in (12). The resultant optimization problem is

$$\begin{aligned} \min_{\mathbf{z}} \; & F(\mathbf{z}) = \sum_{i=1}^{d} \bar{F}_i(z_i) \\ \text{s.t.} \; & z_1 \geq z_2 \geq \cdots \geq z_d \geq 0 \end{aligned} \tag{13}$$

where $\bar{F}_i(z_i) = (z_i - a_i)z_i + w_i z_i$ and $w_i = (2/L)(\lambda_1 + \lambda_2 (d - i))$.

*Remark 1:* This is a standard QP and can be solved by an off-the-shelf QP solver. However, as will be shown in Section III-B, an efficient closed-form solution for the optimal $\mathbf{z}^*$ can be obtained without the need of a numerical solver.

*2) Common Value in Groups:* The following gives a formal definition of "groups."

*Definition 1:* Given $\{z_1, \ldots, z_d\}$, the set of indices $I_{s:t} = \{i \in \mathbb{Z} : s \leq i \leq t\}$, where $s, t \in \mathbb{Z}$, is called a *group* (denoted $\mathcal{G}_{s:t}$) if:

a) $z_i = z_j$ for all $i, j \in I_{s:t}$;
b) $z_{s-1} \neq z_s$ if $s \neq 1$;
c) $z_t \neq z_{t+1}$ if $t \neq d$.

*Proposition 3:* For a group $\mathcal{G}_{s:t}$ in the optimal $\mathbf{z}^*$, the common value for each element in $\mathcal{G}_{s:t}$ is given by

$$z = z_s = \cdots = z_t = \max\{v_{s:t}, 0\} \tag{14}$$

where

$$v_{s:t} \equiv \frac{\sum_{i \in \mathcal{G}_{s:t}} (a_i - w_i)}{2(t - s + 1)}. \tag{15}$$

*3) Coherence:* Another important property is that the groups in $\mathbf{z}^*$ must be *coherent* in the following sense.

*Definition 2:* A group $\mathcal{G}_{s:t}$ is *coherent* if there is no integer $u \in \{s, s+1, \ldots, t-1\}$ such that $v_{s:u} > v_{u+1:t}$.

Intuitively, one cannot split a coherent group into two such that their group values are in descending order.

*Proposition 4:* For any group $\mathcal{G}_{s:t}$ in $\mathbf{z}^*$, if its common value $z_s^* = \cdots = z_t^* = v_{s:t} \geq 0$, then $\mathcal{G}_{s:t}$ is coherent.

Now we are ready to give the sufficient and necessary conditions for the optimal $\mathbf{z}^*$.

**Algorithm 2** Solving of the Proximal Step [Here, $v(\mathcal{G})$ of a Group $\mathcal{G}$ Denotes its Group Value as Defined in (15)]

Compute **a** in (9), and sort the $\{a_i\}_{i=1}^d$ in decreasing order.
**for** $i = 1, 2, \ldots, d$ **do**
  define group $\mathcal{G}_{i:i} = \{i\}$.
**end for**
Initialize stack $\mathcal{S} = \{\mathcal{G}_{1:1}\}$, and let $\mathcal{G}_{\text{top}}$ be the group at the top of $\mathcal{S}$.
**for** $i = 2, 3, \ldots, d$ **do**
  $\bar{\mathcal{G}} \leftarrow \mathcal{G}_{i:i}$.
  **while** $\mathcal{S}$ is not empty and $v(\bar{\mathcal{G}}) \geq v(\mathcal{G}_{\text{top}})$ **do**
    $\bar{\mathcal{G}} \leftarrow \bar{\mathcal{G}} \cup \mathcal{G}_{\text{top}}$. {merge $\bar{\mathcal{G}}$ with $\mathcal{G}_{\text{top}}$}
    remove $\mathcal{G}_{\text{top}}$ from the top of $\mathcal{S}$. {pop}
  **end while**
  add $\bar{\mathcal{G}}$ to the top of $\mathcal{S}$. {push}
**end for**
Compute $\mathbf{z}^*$ from the obtained groups by (14), (15).

*Theorem 1:* $\mathbf{z}^*$ is the optimal solution of (13) if and only if it satisfies Propositions 2–4.

### B. Algorithm

Algorithm 2 shows how to obtain the optimal $\mathbf{z}^*$. Initially, every index $s \in \{1, 2, \ldots, d\}$ forms a group $\mathcal{G}_{s:s} = \{s\}$ of its own. Starting from the first group, the algorithm examines the group values of two consecutive groups. If they are not in decreasing order, the groups are merged and pushed to the stack. The following proposition shows that the merged group is still coherent.

*Proposition 5:* Given two consecutive coherent groups $\mathcal{G}_{s:u}$ and $\mathcal{G}_{u+1,t}$. If $v_{s:u} \leq v_{u+1:t}$, the merged group $\mathcal{G}_{s:t}$ is also coherent.

Hence, when Algorithm 2 terminates, the groups are arranged in decreasing order (and so are $z_1^*, z_2^*, \ldots, z_d^*$), and thus satisfies Proposition 2. After obtaining the group structure, the common value for entries in each group of $\mathbf{z}^*$ is simply the group value given by (14), and thus Proposition 3 is also satisfied. Moreover, Proposition 5 ensures that the merged group is coherent, thus satisfying Proposition 4.

After obtaining $\mathbf{z}^*$, the magnitude of the optimal $\boldsymbol{\beta}^*$ can be recovered from (8) as $|\beta_i^*| = z_i^*$, while its sign is chosen as $\text{sign}(\beta_i) = \text{sign}(L\hat{\beta}_i^t - [\nabla f(\hat{\boldsymbol{\beta}}^t)]_i)$ as shown in the Appendix.

Since there are $d$ groups initially and each merge operation reduces the number of groups by one, there are at most $d - 1$ merge operations. As each merge operation and other stack operations take $O(1)$ time, the complexity of Algorithm 2 is dominated by the initial sorting of $a_i$s, which takes $O(d \log d)$ time. This is much faster than the QP-based and SQP-based solvers in Section II-A, and the network flow algorithms for general structured sparse models in Section II-C.

The number of iterations for FISTA to obtain an $\epsilon$-optimal solution is $O(1/\sqrt{\epsilon})$ [20]. The time to compute the gradient of $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ is usually $O(nd)$. Hence, the total time is $O((1/\sqrt{\epsilon})(d(n + \log d))$. Typically, $n \gg \log d$, and the time thus scales linearly w.r.t. $d$.

**Algorithm 3** Computing the Duality Gap for a Given $\boldsymbol{\beta}$

1: Compute $\tilde{\boldsymbol{\gamma}} = 2\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$.
2: Sort $|\gamma_i|$s in decreasing order and store it as $\boldsymbol{\gamma}$.
3: Compute $r^*(\mathbf{X}^T \nabla \tilde{f}(\mathbf{X}\boldsymbol{\beta}))$ using Proposition 6, and subsequently $\boldsymbol{\alpha}(\boldsymbol{\beta})$ in (17).
4: Output the duality gap in (16).

### C. Computing the Duality Gap

The duality gap is often used as a stopping criterion in convex optimization. To compute the duality gap, first notice that $r(\cdot)$ in (6) is a norm [18]. The dual of OSCAR is then [25]: $\max_{\boldsymbol{\alpha}} -\tilde{f}^*(\boldsymbol{\alpha})$: $r^*(\mathbf{X}^T\boldsymbol{\alpha}) \leq 1$, where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the dual variable, $r^*(\boldsymbol{\gamma}) = \max_{r(\boldsymbol{\beta})\leq 1} \boldsymbol{\gamma}^T\boldsymbol{\beta}$ is the dual norm of $r(\cdot)$, and $\tilde{f}^*(\boldsymbol{\alpha})$ is the Fenchel conjugate[2] of $\tilde{f}(\boldsymbol{\theta}) = \|\mathbf{y} - \boldsymbol{\theta}\|^2$, which can be easily shown to be $\tilde{f}^*(\boldsymbol{\alpha}) = (1/4)\|\boldsymbol{\alpha}\|^2 + \boldsymbol{\alpha}^T\mathbf{y}$. For a solution $\boldsymbol{\beta}$, the duality gap is given by [25]

$$f(\boldsymbol{\beta}) + r(\boldsymbol{\beta}) + \tilde{f}^*(\boldsymbol{\alpha}(\boldsymbol{\beta})) \tag{16}$$

where

$$\boldsymbol{\alpha}(\boldsymbol{\beta}) = \min\left\{1, \frac{1}{r^*(\mathbf{X}^T\nabla\tilde{f}(\mathbf{X}\boldsymbol{\beta}))}\right\} \nabla\tilde{f}(\mathbf{X}\boldsymbol{\beta}) \tag{17}$$

and $\nabla\tilde{f}(\mathbf{X}\boldsymbol{\beta}) = 2(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$. To compute $r^*(\boldsymbol{\gamma})$ in (17), we assume that the indices have been sorted as $|\gamma_1| \geq \cdots \geq |\gamma_d|$. The following proposition shows that $r^*(\boldsymbol{\gamma})$ can then be easily computed.

*Proposition 6:* $r^*(\boldsymbol{\gamma}) = \max_{j=1,\ldots,d}(\sum_{i=1}^{j}|\gamma_i|)/(\sum_{i=1}^{j}\lambda_1 + (d-i)\lambda_2)$.

Algorithm 3 shows the procedure for computing the duality gap. The most expensive step is in the sorting of $|\gamma_i|$s, which takes $O(d\log(d))$ time.

## IV. Experiments

### A. Efficiency of the Proposed Solver

In this section, we first demonstrate the efficiency of the proposed OSCAR solver, which will be called FastOSCAR. We use the regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}^* \in \mathbb{R}^d$, and $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. Following [15], we employ the five synthetic problems in Section IV. These datasets have been popularly used (e.g., [6], [7]) and represent various possible real-world scenarios. Data matrices for the first four are generated as $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, where $\mathbf{C} = [c_{ij}]$ is the covariance matrix. For the fifth data, the inputs are generated as

$$\begin{aligned}
x_i &= Z_1 + \epsilon_i,\ Z_1 \sim \mathcal{N}(0,1) \quad i = 1,\ldots,0.1d\\
x_i &= Z_2 + \epsilon_i,\ Z_2 \sim \mathcal{N}(0,1) \quad i = 0.1d+1,\ldots,0.2d\\
x_i &= Z_3 + \epsilon_i,\ Z_3 \sim \mathcal{N}(0,1) \quad i = 0.2d+1,\ldots,0.3d\\
x_i &\sim \mathcal{N}(0,1) \quad i = 0.3d+1,\ldots,d
\end{aligned}$$

where $\epsilon_i \sim \mathcal{N}(0, 0.16)$. Values of $\sigma, \mathbf{C}$, and $\boldsymbol{\beta}^*$ for the five problems are given as follows.

1) $\sigma = 3$, $c_{ij} = 0.7^{|i-j|}$ and $\boldsymbol{\beta}^* = [\underbrace{3,3,\ldots}_{0.1d}, \underbrace{2,2,\ldots}_{0.1d}, \underbrace{1.5,1.5,\ldots}_{0.1d}, \underbrace{0,0,\ldots}_{0.7d}]^T$.
2) Same as dataset 1 except that $\boldsymbol{\beta}^* = [\underbrace{3,3,\ldots}_{0.1d}, \underbrace{0,0,\ldots}_{0.3d}, \underbrace{1.5,1.5,\ldots}_{0.1d}, \underbrace{0,0,\ldots}_{0.4d}, \underbrace{2,2,\ldots}_{0.1d}]^T$.
3) Same as dataset 1 except that $\boldsymbol{\beta}^* = [0.85, 0.85, \ldots, 0.85]^T$.
4) $\sigma = 15$, $c_{ij} = 0.5$ when $i \neq j$, and 1 otherwise; and $\boldsymbol{\beta}^* = [\underbrace{0,\ldots,0}_{0.3d}, \underbrace{2,\ldots,2}_{0.2d}, \underbrace{0,\ldots,0}_{0.3d}, \underbrace{2,\ldots,2}_{0.2d}]^T$.
5) $\sigma = 15$ and $\boldsymbol{\beta}^* = [\underbrace{3,\ldots,3}_{0.3d}, \underbrace{0,\ldots,0}_{0.7d}]^T$.

[2]The Fenchel conjugate of $\tilde{f}(\boldsymbol{\theta})$ is defined as $\tilde{f}^*(\boldsymbol{\alpha}) = \sup_{\boldsymbol{\theta}} \boldsymbol{\alpha}^T\boldsymbol{\theta} - \tilde{f}(\boldsymbol{\theta})$.

TABLE I
Values of the Exponent $\eta$ in the Empirical Time Complexities $O(d^\eta)$ on the Five Synthetic Datasets

| | datasets | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| QP | 4.35 | 4.26 | 5.22 | 4.49 | 2.78 |
| SQP | 3.50 | 3.20 | 3.73 | 2.56 | 2.74 |
| ProxFlow | 2.38 | 2.34 | 2.38 | 2.47 | 2.03 |
| FastOSCAR | 1.75 | 1.74 | 1.64 | 0.94 | 1.00 |

We compare FastOSCAR with: 1) the QP solver in [15]; 2) the SQP solver[3] in [15]; and 3) ProxFlow[4] in [19]. For the QP and SQP solvers, the default stopping criterion is used. For ProxFlow, the default maximum number of iterations is 100. However, this cannot yield a comparable accuracy to the other solvers here, and so we increase it to 300. For FastOSCAR, it stops when the relative duality gap is small (i.e., $(f(\boldsymbol{\beta}) + r(\boldsymbol{\beta}) + f^*(\boldsymbol{\alpha}(\boldsymbol{\beta})))/(f(\boldsymbol{\beta}) + r(\boldsymbol{\beta})) \leq 10^{-6}$) or when the relative change in the OSCAR objective is small ($\leq 10^{-5}$), with a maximum of 2000 iterations.

We vary the input dimensionality $d$ from 10 to 10 240. For each $d$, $\lambda_1$ and $\lambda_2$ are chosen from a validation set of 1000 samples. [15, eq. (7)] is then used to obtain the equivalent $(c, t)$ setting for the QP and SQP solvers. Reported results are averaged from ten realizations of the training sets (each with 1000 samples). Experiments are performed on a PC with a quad-core 4.5-GHz CPU and 12-GB memory.

On all the runs, the four solvers return almost identical solutions (the relative differences in the obtained objective values and $\|\boldsymbol{\beta}\|$ are typically smaller than $10^{-5}$). Hence, only the time is reported in Fig. 2. As can be seen, FastOSCAR is much faster than the others even when $d$ is only moderately large (e.g., $d \geq 80$). The empirical time complexities are shown in Table I. The running time of FastOSCAR is close to $O(d)$, which agrees with our analysis in Section III-B.

Next, we fix $n$ and $d$ to 1000 and study the effect of $\lambda_1, \lambda_2$ on the speed of FastOSCAR. Because of the lack of space, only the results on dataset 5 are shown in Fig. 3. As can be seen, it is not sensitive to the value of $\lambda_1$. As for $\lambda_2$, when it is large, the grouping effect is strong and a few large clusters can be quickly formed. Hence, the number of FISTA iterations is small [Fig. 3(a)] and so is the CPU time [Fig. 3(b)].

Finally, we set $n = d$ and vary $d$ from 1 to 2560. Fig. 4 shows the speed of FastOSCAR. As can be seen, it

[3]Both the QP and SQP solvers are available at: http://www4.stat.ncsu.edu/~bondell/software.html.

[4]Available at: http://www.di.ens.fr/~mairal/software.php.

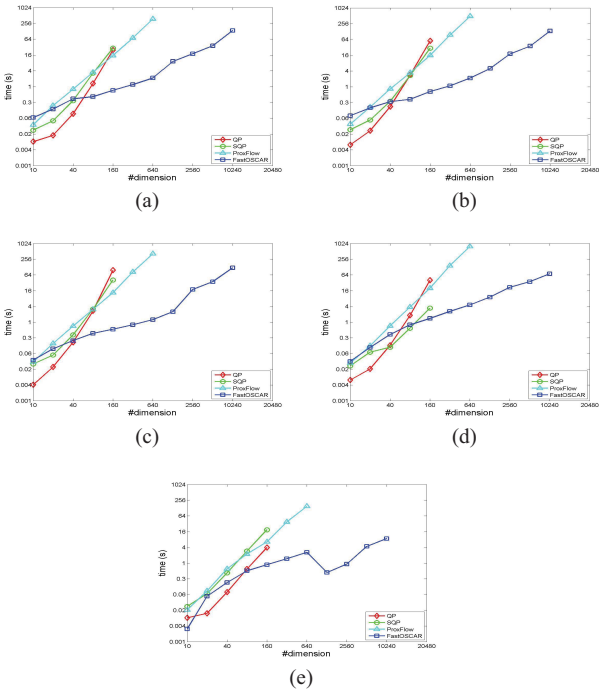



Fig. 2. CPU time for various OSCAR solvers on the synthetic datasets. Note that both axes are in log scale. (a) Data set 1. (b) Data set 2. (c) Data set 3. (d) Data set 4. (e) Data set 5.

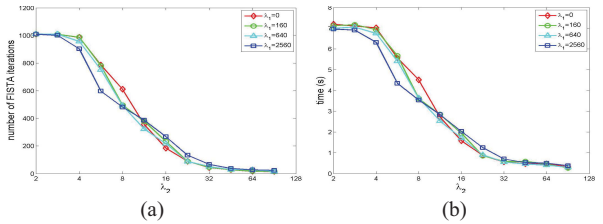



Fig. 3. Speed of FastOSCAR on dataset 5, at different values of $\lambda_1$ and $\lambda_2$. (a) Number of FISTA iterations. (b) CPU time.

is almost linear in $nd$, which again agrees with our analysis in Section III-B.

### B. Synthetic Data

In this section, we compare OSCAR with several state-of-the-art sparse-modeling approaches, including lasso [6], fused lasso [14], elastic net [7], and group lasso [11]. The datasets are generated as in Section IV-A except with:

1) $d = 8$ and $\boldsymbol{\beta}^* = [3, 2, 1.5, 0, 0, 0, 0, 0]^T$;
2) $d = 8$ and $\boldsymbol{\beta}^* = [3, 0, 0, 1.5, 0, 0, 0, 2]^T$;
3) $d = 8$ and $\boldsymbol{\beta}^* = [0.85, 0.85, \ldots, 0.85]^T$;
4) $d = 40$ and $\boldsymbol{\beta}^* = [\underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}, \underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}]^T$;
5) $d = 40$ with $\boldsymbol{\beta}^* = [\underbrace{3, \ldots, 3}_{5}, \underbrace{3, \ldots, 3}_{5}, \underbrace{, 3 \ldots, 3}_{5}, \underbrace{0, \ldots, 0}_{25}]^T$.

A summary of the datasets is given in Table II. As suggested in [14], feature ordering for the fused lasso is obtained by hierarchical clustering. Group lasso requires the group structure to be known in advance. In this experiment, we give it an unfair advantage by supplying it with the true group structure. Parameters in all the models are tuned using an independent validation set of the same size as the training set.

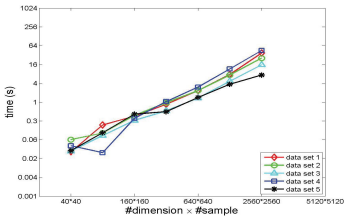



Fig. 4. CPU time of FastOSCAR at different values of $nd$.

TABLE II
SUMMARY OF THE DATASETS USED

| Data | Dim | No. of training | No. of validation | No. of test |
|---|---|---|---|---|
| Synthetic problem 1 | 8 | 40 | 40 | 200 |
| Synthetic problem 2 | 8 | 40 | 40 | 200 |
| Synthetic problem 3 | 8 | 40 | 40 | 200 |
| Synthetic problem 4 | 40 | 200 | 200 | 400 |
| Synthetic problem 5 | 40 | 100 | 100 | 400 |
| atheism versus graphics | 7971 | 707 | 707 | 354 |
| windows.x versus religion.misc | 8172 | 644 | 643 | 322 |
| autos versus motorcycles | 8012 | 792 | 792 | 396 |
| baseball versus hockey | 8087 | 795 | 795 | 398 |
| forsale versus ms-windows.misc | 6818 | 740 | 739 | 370 |
| guns versus mideast | 10 091 | 789 | 789 | 394 |
| med versus space | 9510 | 771 | 771 | 385 |
| pc.hardware versus politics.misc | 8196 | 701 | 701 | 351 |
| mac.hardware versus christian | 8257 | 782 | 782 | 391 |
| crypt versus electronics | 8778 | 789 | 789 | 395 |
| breast cancer | 300 | 118 | 118 | 59 |

As discussed in [8], encouraging coefficients to be similar to each other as in OSCAR will unwantedly overpenalize large pairwise coefficient differences and thus impede performance. Instead of resorting to the use of a nonconvex regularizer as in [8], we alleviate this problem by rescaling OSCAR. First, we run OSCAR to obtain the group structure. Features with zero coefficients are discarded, while those in the same group $\mathcal{G}$ are merged to form a new "super-feature" $\mathbf{x}_{\mathcal{G}} = \sum_{i \in \mathcal{G}} \operatorname{sgn}(\beta_i)\mathbf{x}_i$ of weight $|\mathcal{G}|$. These are then used to train a weighted ridge regression model. For comparison, this rescaling is also applied to the fused lasso.

Table III compares the various methods in terms of the following:

1) mean-squared error (MSE) $(\boldsymbol{\beta} - \boldsymbol{\beta}^*)\mathbf{X}^T\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)$;
2) degrees of freedom (DoF), which is the number of unique nonzero coefficients obtained, for OSCAR, it is the number of groups discovered [15];
3) number of feature selection errors, which is the number of nonzero coefficients that are estimated as zero plus the number of zero coefficients that are estimated as nonzero.

TABLE III

RESULTS ON THE SYNTHETIC DATA. NUMBERS AFTER THE ± SIGN ARE THE STANDARD ERRORS OF THE MEDIAN (ESTIMATED BY BOOTSTRAPPING WITH 500 RESAMPLINGS AS IN [7]). THE BEST RESULTS AND THOSE THAT ARE NOT SIGNIFICANTLY WORSE (ACCORDING TO THE $t$-TEST WITH A $p$-VALUE LESS THAN 0.05) ARE IN BOLD. NOTE THAT UNLIKE OSCAR, GROUP LASSO IS GIVEN THE TRUE GROUP STRUCTURE

| Dataset | | Lasso | Fused lasso (nonrescaled) | Fused lasso (rescaled) | Elastic net | Group lasso | OSCAR (nonrescaled) | OSCAR (rescaled) |
|---|---|---|---|---|---|---|---|---|
| 1 | MSE | 0.93±0.25 | 0.69±0.10 | 0.38±0.07 | **0.33±0.08** | 1.17±0.23 | 0.79±0.19 | **0.33±0.08** |
| | DoF | 4.73±1.14 | 3.97±1.51 | **2.48±0.69** | 3.84±0.73 | 7.46±1.32 | 4.89±1.22 | **2.48±0.71** |
| | No. of (selection error) | 2.08±1.34 | 2.76±1.85 | **0.39±0.72** | 0.84±0.73 | 4.82±1.37 | 2.94±1.50 | **0.42±1.39** |
| 2 | MSE | 1.16±0.14 | 1.46±0.20 | 1.13±0.17 | 1.06±0.13 | 1.37±0.16 | 1.13±0.13 | **0.98±0.15** |
| | DoF | 5.11±1.20 | 5.64±1.04 | **4.84±1.10** | 6.10±1.05 | 7.50±1.00 | 5.50±0.89 | 5.24±1.69 |
| | No. of (selection error) | **2.12±1.21** | 4.35±1.00 | 3.41±1.39 | 3.12±1.05 | 4.51±1.00 | 3.13±1.08 | 3.71±1.44 |
| 3 | MSE | 1.61±0.09 | **0.02±0.02** | **0.02±0.02** | 0.87±0.10 | 0.79±0.16 | 0.48±0.21 | **0.02±0.02** |
| | DoF | 6.31±0.88 | **1.12±0.33** | **1.12±0.32** | 7.60±0.52 | 8.00±0.00 | 3.54±1.26 | **1.12±0.32** |
| | No. of (selection error) | 1.69±0.88 | **0.00±0.00** | **0.00±0.00** | 0.40±0.52 | **0.00±0.00** | 0.15±0.36 | **0.00±0.00** |
| 4 | MSE | 26.96±1.06 | 17.61±0.88 | 15.71±0.56 | 21.01±0.78 | **9.73±1.22** | 19.00±1.01 | 16.00±0.75 |
| | DoF | 25.29±3.04 | **7.84±6.62** | 22.29±9.02 | 27.14±2.15 | 31.98±6.60 | 17.68±5.55 | 22.10±11.14 |
| | No. of (selection error) | 12.97±2.44 | 18.55±2.42 | 16.51±3.45 | **11.11±2.04** | 11.98±6.60 | 18.78±1.84 | 16.75±3.74 |
| 5 | MSE | 33.41±2.19 | 12.37±1.32 | 4.17±0.66 | 13.85±1.28 | 22.62±2.51 | 23.58±2.10 | **3.84±1.22** |
| | DoF | 13.86±3.64 | 7.09±2.46 | 4.03±1.07 | 15.78±1.37 | 40.00±0.00 | 18.19±4.06 | **2.82±1.30** |
| | No. of (selection error) | 9.04±3.63 | 7.71±6.60 | 0.62±1.40 | 1.61±0.82 | 25.00±0.00 | 14.11±4.82 | **0.00±0.00** |

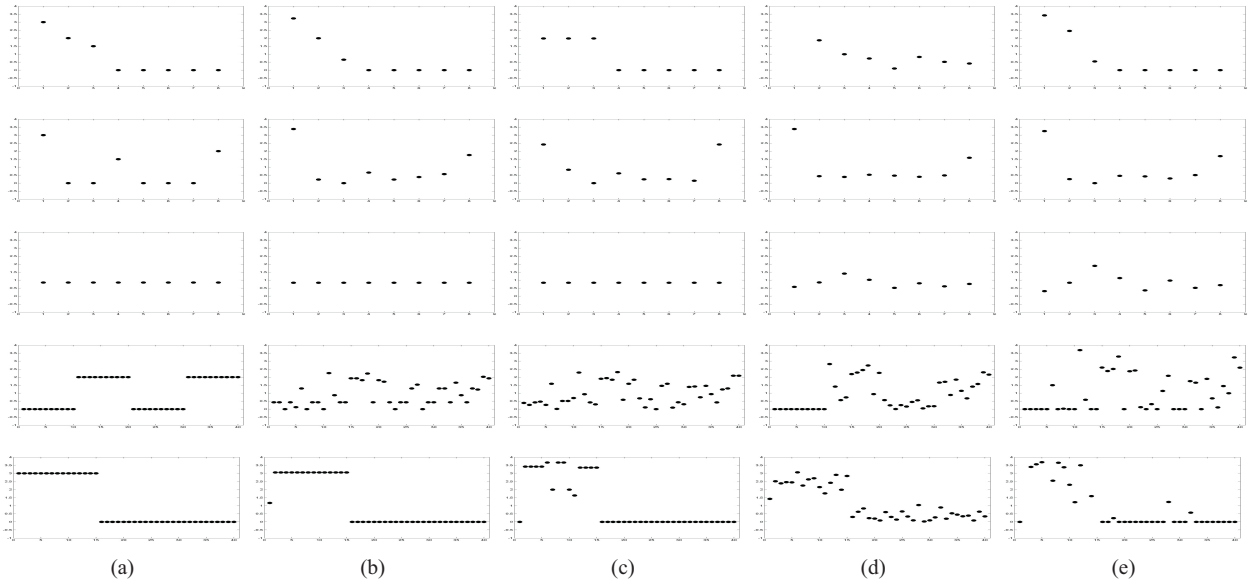



Fig. 5. Feature coefficients obtained on the five datasets. The first row is for dataset 1, the second row is for dataset 2, and so on. (a) Ground truth. (b) OSCAR. (c) Fused lasso. (d) Group lasso. (e) Elastic net.

To reduce statistical variability, results for all the methods are averaged over 50 repetitions.

As can be seen, rescaling helps both OSCAR and fused lasso. Overall, (rescaled) OSCAR excels on all three criteria. By automatically tying features into groups, OSCAR is able to reduce the DoF and thus model complexity, making it less prone to overfitting and achieving the best or second-best MSE among all methods. Moreover, it has a much lower feature selection error. The rescaled fused lasso is also sometimes competitive, but OSCAR is advantageous in that the features do not need to be first ordered by an additional hierarchical clustering step. Group lasso, though given the true group structure, is often inferior to OSCAR. We speculate that this is because group lasso does not tie the nonzero coefficients in the same group together, and thus has a much higher complexity, which impairs generalization.

Fig. 5 shows the feature coefficients obtained by the various methods in a typical run. As can be seen, among the sparse-modeling methods tested, only OSCAR and fused lasso can perform feature grouping. Moreover, on comparing with the fused lasso, OSCAR can group most of the zero features together and its solution is also closer to the ground truth.

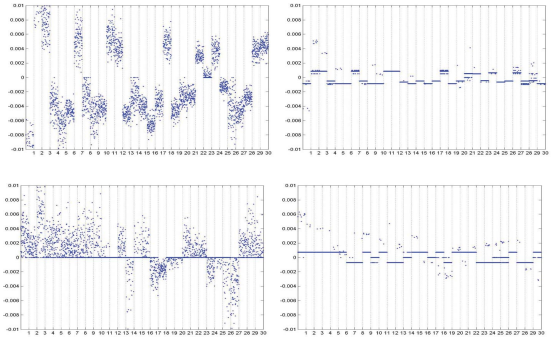
### C. 20 Newsgroups

In this experiment, we divide the 20-newsgroups dataset[5] into ten pairs (Table II). As preprocessing, we remove words that appear in fewer than three documents. 40% of the samples are then used for training, another 40% for validation, and the

[5]Available at: http://people.csail.mit.edu/jrennie/20Newsgroups/.

TABLE IV
RESULTS ON THE 20-NEWSGROUPS SUBSET. THE BEST RESULTS AND THOSE THAT ARE NOT SIGNIFICANTLY WORSE (ACCORDING TO THE $t$-TEST WITH A $p$-VALUE LESS THAN 0.05) ARE IN BOLD

| Dataset | | Lasso | Fused lasso (nonrescaled) | Fused lasso (rescaled) | Elastic net | OSCAR (nonrescaled) | OSCAR (rescaled) |
|---|---|---|---|---|---|---|---|
| atheism | Test accuracy | 92.09±0.98 | 93.11±0.84 | 93.28±1.10 | **94.07±1.26** | **93.16±1.50** | **93.95±1.36** |
| versus | DoF | **445±81** | **444±79** | **964±584** | 4803±1312 | **576±164** | **1094±1130** |
| graphics | No. of nonzero feat. | **467±87** | 3165±1072 | 2544±1997 | 4963±1350 | 6016±1787 | 4678±2866 |
| windows.x | Test accuracy | 91.68±1.06 | 93.04±0.84 | 93.60±0.92 | **94.91±1.35** | 93.42±1.94 | 94.04±0.86 |
| versus | DoF | **437±65** | **459±74** | 692±267 | 5259±1528 | **445±159** | **562±214** |
| religion.misc | No. of nonzero feat. | 457±74 | 2796±1285 | 2465±1477 | 5486±1590 | 6279±1584 | 6349±1269 |
| autos | Test accuracy | 90.45±0.98 | 92.47±1.27 | **93.54±1.50** | **94.19±1.18** | 92.17±1.43 | **93.99±1.12** |
| versus | DoF | **419±73** | **383±108** | **603±232** | 6027±1104 | **359±181** | 779±285 |
| motorcycles | No. of nonzero feat. | **438±76** | 3541±818 | 4533±2415 | 6325±1178 | 6749±551 | 5658±2731 |
| baseball | Test accuracy | 89.95±2.42 | 91.56±2.01 | **92.66±0.93** | **93.52±1.47** | **92.66±1.41** | **93.47±1.19** |
| versus | DoF | **332±97** | **368±89** | 967±485 | 6326±903 | **238±110** | 660±378 |
| hockey | No. of nonzero feat. | **342±102** | 3217±1254 | 3238±2043 | 6583±953 | 6778±1048 | 5994±1223 |
| forsale | Test accuracy | 93.14±2.37 | 94.65±1.20 | 96.05±0.68 | **96.70±0.67** | **94.70±2.31** | **95.30±1.59** |
| versus | DoF | **389±93** | **361±34** | 802±206 | 5663±1903 | **290±151** | **764±485** |
| ms-windows.misc | No. of nonzero feat. | **416±98** | 2181±1098 | 5186±2258 | 5903±2008 | 4116±688 | 5001±343 |
| guns | Test accuracy | 89.49±1.67 | 91.12±1.26 | 92.03±2.19 | **93.20±1.54** | 90.91±1.93 | 92.18±1.51 |
| versus | DoF | **402±126** | 500±89 | 1118±629 | 6052±1935 | **349±109** | 705±166 |
| mideast | No. of nonzero feat. | **420±139** | 3220±532 | 4353±2367 | 6308±2016 | 4435±866 | 5415±1199 |
| med | Test accuracy | 91.32±1.50 | 91.74±2.25 | **93.35±1.25** | **93.40±1.62** | **92.57±1.67** | **93.35±1.73** |
| versus | DoF | **315±13** | **363±135** | 691±234 | 4600±1364 | **201±141** | 736±288 |
| space | No. of nonzero feat. | **323±16** | 3210±1527 | 3510±2414 | 4714±1425 | 4707±1110 | **2680±2529** |
| pc.hardware | Test accuracy | 94.07±1.11 | 95.44±1.72 | **96.24±0.74** | **96.98±0.91** | **95.56±1.18** | **96.41±1.25** |
| versus | DoF | **286±76** | 394±106 | **531±294** | 3965±782 | **327±85** | **425±214** |
| politics.misc | No. of nonzero feat. | 294±79 | 2757±1739 | 4038±2215 | 4095±799 | 4435±517 | 5127±2635 |
| mac.hardware | Test accuracy | 93.81±1.44 | **94.58±1.22** | **95.04±0.92** | **95.14±1.27** | **95.19±1.47** | **95.40±1.49** |
| versus | DoF | **267±155** | 433±55 | 629±238 | 5793±1185 | **260±133** | **345±76** |
| christian | No. of nonzero feat. | **274±163** | 2760±1542 | 2858±1910 | 5973±1241 | 5367±970 | 3964±3261 |
| crypt | Test accuracy | 88.56±0.90 | 90.94±1.87 | 90.68±0.85 | **92.46±1.03** | 90.68±0.55 | 91.24±0.73 |
| versus | DoF | **453±83** | **455±60** | **891±683** | 5886±2398 | **444±132** | **740±306** |
| electronics | No. of nonzero feat. | **478±92** | 3989±646 | 4802±1929 | 6151±2515 | 5421±620 | 3512±2557 |



(a) (b)

Fig. 6. Coefficients for weights in the additional groups of "autos versus motorcycles" (top) and "windows.x versus religion.misc" (bottom). The 30 groups are delineated by vertical lines. (a) Elastic net. (b) OSCAR.

remaining 20% for testing. To make the dataset more challenging, we first run ridge regression, and then duplicate the 30 most important features 100 times (with noise from $\mathcal{N}(0, 0.16)$ added to the nonzero features) to form 30 additional groups. The prediction on a test sample $\mathbf{x}$ is $\text{sign}(\mathbf{x}^T\boldsymbol{\beta})$.

Table IV shows the classification accuracies, DoF, and the number of feature selection errors (averaged over five repetitions). As can be seen, while the elastic net is the most accurate, its DoF is much larger. In contrast, lasso and fused lasso often have small DoFs, but their accuracies are inferior. On the other hand, the accuracy of OSCAR is comparable with that of the elastic net, and yet enjoys a small DoF. Moreover, note that the nonrescaled OSCAR is often as good as its rescaled version. We speculate that since feature grouping is strong on this dataset, the OSCAR model well matches the data. Hence, overpenalization of the coefficients is much less a problem, making rescaling less important.

Fig. 6 compares the coefficients obtained by the elastic net and OSCAR on the 30 additional groups of "atheism versus graphics" and "windows.x versus religion.misc" (results for the other pairs are similar). Though the elastic net has some grouping effect, it fails to tie together features in the same group (which are duplicates of each other). In contrast, OSCAR produces much clearer feature groups. This is also consistent with the results on the synthetic datasets in Fig. 5.

### D. Breast Cancer

In this section, we experiment with the breast cancer dataset,[6] which contains 8141 genes in 295 tumors. We use the 300 genes that are most correlated with the output, and reduce the class imbalance by duplicating the positive samples twice. 40%, 40%, and 20% of the dataset are then randomly chosen

[6]Available at: http://cbio.ensmp.fr/∼ljacob/.

TABLE V
RESULTS ON THE BREAST CANCER DATASET. THE BEST RESULTS AND THOSE THAT ARE NOT SIGNIFICANTLY WORSE (ACCORDING TO THE $t$-TEST WITH A $p$-VALUE LESS THAN 0.05) ARE IN BOLD

| | Lasso | Fused lasso (nonrescaled) | Fused lasso (rescaled) | Elastic net | OSCAR (nonrescaled) | OSCAR (rescaled) |
|---|---|---|---|---|---|---|
| Classification accuracy | 72.49±7.35 | **75.80±3.97** | **76.48±5.25** | **76.20±5.03** | 72.51±7.06 | **77.55±5.57** |
| DoF | 39.8±23.74 | **23.3±15.64** | **32.2±14.07** | 178.1±79.93 | 41.7±31.41 | **38.7±20.70** |
| No. of nonzero features | **39.8±23.74** | 215.8±67.57 | **102.1±78.78** | 178.1±79.93 | **143.0±113.89** | **79.3±81.61** |

for training, validation, and testing, respectively. Table V shows the results averaged over ten repetitions. As can be seen, the (rescaled) OSCAR and fused lasso have comparable accuracy to the elastic net, but with a much smaller DoF.

## V. CONCLUSION

In this paper, we used the accelerated gradient method to solve the optimization problem associated with the structured sparse OSCAR model. We showed that the core proximal step can be solved with an iterative group merging algorithm which is simple, easy to implement, and much more efficient than state-of-the-art solvers. Experimental results on a number of toy and real-world datasets showed that it is a competitive sparse-modeling approach for both regression and classification problems, but with the added ability of automatic feature grouping.

As discussed in Section II-B, the proximal step is a core component in many other proximal and accelerated gradient methods. The proposed efficient computation of the proximal step can also be used with these methods and is not limited to FISTA. Moreover, since the (accelerated) gradient method only requires knowledge of the gradient, the proposed algorithm can be used to extend OSCAR to other loss functions besides the square loss. These will be further investigated in the future.

## APPENDIX

### A. Proof of Proposition 1

Using (5) and (6), the objective $Q(\boldsymbol{\beta}; \hat{\boldsymbol{\beta}}^t)$ in (3) becomes

$$f(\hat{\boldsymbol{\beta}}^t) + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^t)^T \nabla f(\hat{\boldsymbol{\beta}}^t) + \frac{L}{2}\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^t\|^2 + r(\boldsymbol{\beta})$$
$$= \frac{L}{2}\sum_{i=1}^{d}\left(\beta_i^2 - \frac{2(L\hat{\beta}_i^t - \nabla f_i^t)\beta_i}{L}\right) + r(\boldsymbol{\beta}) + C$$

where $\nabla f_i^t = [\nabla f(\hat{\boldsymbol{\beta}}^t)]_i$ and $C = f(\hat{\boldsymbol{\beta}}^t) - \nabla f(\hat{\boldsymbol{\beta}}^t)^T \hat{\boldsymbol{\beta}}^t + (L/2)\|\hat{\boldsymbol{\beta}}^t\|^2$. Since $C$ is a constant, this is also equivalent to

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{d}\left(\beta_i^2 - \frac{2(L\hat{\beta}_i^t - \nabla f_i^t)\beta_i}{L}\right) + \frac{2}{L}r(\boldsymbol{\beta}). \quad (18)$$

The only term that depends on the sign of $\beta_i$ is $(L\hat{\beta}_i^t - \nabla f_i^t)\beta_i$. To minimize (18), the sign of each $\beta_i$ has to be chosen as $\text{sign}(\beta_i) = \text{sign}(L\hat{\beta}_i^t - \nabla f_i^t)$. The result then follows using the definitions in (8) and (9).

### B. Proof of Proposition 2

Obviously, all the $z_i^*$s are $\geq 0$ because of the constraint in (7). Assume to the contrary that $z_j^* > z_i^*$ for some $j > i$. First, consider the case where all the inequalities in (11) are strict inequalities. Construct a new $\mathbf{z}'$ from $\mathbf{z}^*$ by exchanging the values of $z_j^*$ and $z_i^*$. Obviously, $\hat{r}(\mathbf{z}^*) = \hat{r}(\mathbf{z}')$. Together with the fact that $(z_j^* - a_j)z_j^* + (z_i^* - a_i)z_i^* > (z_j^* - a_i)z_j^* + (z_i^* - a_j)z_i^*$, we have $F(\mathbf{z}') < F(\mathbf{z}^*)$, which is a contradiction.

Now consider the case[7] where some $a_i$s are equal, i.e., $a_{i-1} > a_i = a_{i+1} = \cdots = a_j > a_{j+1}$ for some $1 \leq i \leq j \leq d$. Assume to the contrary that some of $\{z_i^*, z_{i+1}^*, \ldots, z_j^*\}$ are not equal. We construct a new $\mathbf{z}'$ from $\mathbf{z}^*$ by exchanging the values of $z_{k_1}$ and $z_{k_2}$, where $i \leq k_1 < k_2 \leq j$. It is easy to see that $F(\mathbf{z}^*) = F(\mathbf{z}')$. Hence, $\mathbf{z}'$ is also an optimal solution, contradicting Corollary 1.

### C. Proof of Proposition 3

First, we introduce the following lemma.

*Lemma 1:* Given a particular group $\mathcal{G}_{s:t}$, the subsum $\sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z_i)$: $z_s \geq \ldots \geq z_t \geq 0$ in (13) is minimized by the *group value* in (14).

*Proof:* As $z_s = z_{s+1} = \cdots = z_t$, the objective reduces to the univariate convex quadratic function $\sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z)$, whose minimum can be easily obtained as (14). ■

We now proceed with the proof of Proposition 3. Assume to the contrary that there is a group $\mathcal{G}_{s:t}$ with $z_s^* = \cdots = z_t^* \neq \max\{v_{s:t}, 0\}$. Construct $\mathbf{z}'$ from $\mathbf{z}^*$ as

$$z_s' = \cdots = z_t' = \min\{z_{s-1}^*, \max\{v_{s:t}, z_{t+1}^*\}\}^8 \quad (19)$$

while all the other entries in $\mathbf{z}'$ are the same as those in $\mathbf{z}^*$. Since $\mathbf{z}^*$ satisfies Proposition 2, clearly so does $\mathbf{z}'$. Consider the following three cases.

1) $v_{s:t} > z_{s-1}^*$: from the definition of groups and Proposition 2

$$z_{s-1}^* > z_s^* = \cdots = z_t^* > z_{t+1}^*. \quad (20)$$

Hence, $v_{s:t} > z_{s-1}^* > z_s^* > z_{t+1}^*$ (Fig. 7). Moreover, as $z_{t+1}^* \geq 0$, hence $v_{s:t} > 0$ and thus $\arg\min \sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z) = \max\{v_{s:t}, 0\} = v_{s:t}$. From (19), $z_s' = z_{s-1}^*$, which is then closer than $z_s^*$ to $v_{s:t}$. Since $\sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z)$ is quadratic, $\sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z_s') < \sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z_s^*)$, and thus $F(\mathbf{z}') < F(\mathbf{z}^*)$, a contradiction.

[7] The borderline cases "$i = 1$" and "$j = d$" can be easily handled and are not discussed here.

[8] Here, we assume $z_0^* = \infty$ and $z_{d+1}^* = 0$.

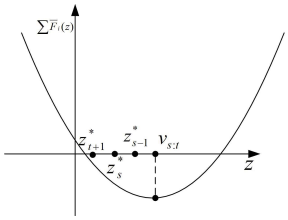



Fig. 7. Positions of $v_{s:t}, z^*_{s-1}, z^*_s, z^*_{t+1}$ in the proof of Case 1 of Proposition 3. Note that $z'_s = z^*_{s-1}$.

2) $z^*_{t+1} > v_{s:t}$: similar to Case 1, we have $\sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z'_s) < \sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z^*_s)$ and thus $F(\mathbf{z}') < F(\mathbf{z}^*)$, a contradiction.
3) $z^*_{s-1} \geq v_{s:t} \geq z^*_{t+1}$: as $z^*_{t+1} \geq 0$ (due to the nonnegative constraint), we have $v_{s:t} \geq 0$, and thus $\arg\min \sum_{i\in\mathcal{G}_{s:t}} \bar{F}_i(z) = v_{s:t}$. Moreover, from (19), $z'_s = \cdots = z'_t = v_{s:t}$, and thus $F(\mathbf{z}') < F(\mathbf{z}^*)$, a contradiction.

### D. Proof of Proposition 4

Before showing the proof of Proposition 4, we first introduce the following lemma.

*Lemma 2:* If $v_{s:u} < v_{u+1:t}$, then $v_{s:u} < v_{s:t} < v_{u+1:t}$. Similarly, if $v_{s:u} > v_{u+1:t}$, then $v_{s:u} > v_{s:t} > v_{u+1:t}$. The inequalities become equalities when $v_{s:u} = v_{u+1:t}$.

*Proof:* Define $p_{s:u} \equiv \sum_{i\in\mathcal{G}_{s:u}}(a_i - w_i)$, $q_{s:u} \equiv 2(u+1-s)$, and similarly $p_{u+1:t}$ and $q_{u+1:t}$. Then $v_{s:u} = (p_{s:u}/q_{s:u})$ and $v_{u+1:t} = (p_{u+1:t}/q_{u+1:t})$. From (15), we also have $v_{s:t} = ((p_{s:u} + p_{u+1:t})/(q_{s:u} + q_{u+1:t})$. Now if $(p_1/q_1) < (p_2/q_2)$ and $q_1, q_2 > 0$, then $(p_1/q_1) < (p_1+p_2)/(q_1+q_2) < (p_2/q_2)$. Hence, the first part of the lemma follows. The proof of the second part is similar. ∎

We now proceed with the proof of Proposition 4. Assume to the contrary that there exists an incoherent group $\mathcal{G}_{s:t}$ with $z^*_s = \cdots = z^*_t = v_{s:t} \geq 0$. In other words, there exists a split at some $u \in \{s, s+1, \ldots, t-1\}$ such that

$$v_{s:u} > v_{u+1:t}. \tag{21}$$

Now construct a new $\mathbf{z}'$ from $\mathbf{z}^*$ by setting

$$z'_s = \cdots = z'_u = \min\{z^*_{s-1}, v_{s:u}\}. \tag{22}$$

$z'_{u+1} = \cdots = z'_t = \max\{z^*_{t+1}, v_{u+1:t}\}$, while the other entries are the same as $\mathbf{z}^*$. Clearly $\mathbf{z}'$ still satisfies Proposition 2. Consider the following two cases.

1) $v_{s:u} \geq z^*_{s-1}$: similar to Case 1 in the proof of Proposition 3, we have $v_{s:u} \geq z^*_{s-1} > z^*_s$. From (22), $z'_s = z^*_{s-1}$, which is closer than $z^*_s$ to $\arg\min \sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z) = \max\{v_{s:u}, 0\} = v_{s:u}$ by (20). Thus, $\sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z'_s) < \sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z^*_s)$.
2) $v_{s:u} < z^*_{s-1}$: it then follows from (22) that

$$z'_s = v_{s:u}. \tag{23}$$

Using (21) and Lemma 2, we have

$$v_{s:u} > v_{s:t}. \tag{24}$$

Since $v_{s:t} \geq 0$ by assumption, we have $v_{s:u} > v_{s:t} \geq 0$. Thus, $\arg\min \sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z) = \max\{v_{s:u}, 0\} = v_{s:u}$. Hence, from (20), (23), and (24), $z'_s = v_{s:u} > v_{s:t} = z^*_s = \cdots = z^*_t > z^*_{t+1}$. Since $v_{s:u}$ is the minimizer of $\sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z)$, so $\sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z'_s) < \sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z^*_s)$.

In both cases, $\sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z'_s) < \sum_{i\in\mathcal{G}_{s:u}} \bar{F}_i(z^*_s)$. Similarly, one can show $\sum_{i\in\mathcal{G}_{u+1:t}} \bar{F}_i(z'_t) \leq \sum_{i\in\mathcal{G}_{u+1:t}} \bar{F}_i(z^*_t)$.[9] Thus, $F(\mathbf{z}') < F(\mathbf{z}^*)$ and $\mathbf{z}^*$ is not optimal, a contradiction.

### E. Proof of Theorem 1

Assume that

$$\exists \mathbf{z}, \hat{\mathbf{z}}\ (\mathbf{z} \neq \hat{\mathbf{z}}) \text{ satisfying Propositions 2–4.} \tag{25}$$

Obviously, their grouping structures are different; otherwise, we have $\mathbf{z} = \hat{\mathbf{z}}$. Suppose that $\mathbf{z}$ and $\hat{\mathbf{z}}$ are equal up to the first $u-1$ $z_i$s

$$z_1 = \hat{z}_1, z_2 = \hat{z}_2, \ldots, z_{u-1} = \hat{z}_{u-1}, \quad \text{but} \quad z_u \neq \hat{z}_u. \tag{26}$$

Let $u$ belong to some (coherent) group $\mathcal{G}_{s:t}$ (resp. $\mathcal{G}_{\hat{s}:\hat{t}}$) in $\mathbf{z}$ (resp. $\hat{\mathbf{z}}$).

*Lemma 3:* $t \neq \hat{t}$.

*Proof:* Assume to the contrary that $t = \hat{t}$. Consider the following cases.

1) $s < \hat{s}$: as $u$ belongs to $\mathcal{G}_{\hat{s}:\hat{t}}$ in $\hat{\mathbf{z}}$ (resp. $\mathcal{G}_{s:t}$ in $\mathbf{z}$), we have $s < \hat{s} \leq u$. By (26) and definition 1, $z_{s-1} = \hat{z}_{s-1} > z_s = \hat{z}_s = z_{s+1} = \hat{z}_{s+1} \cdots = z_u > \hat{z}_u$, so $u$ is the start of some group in $\hat{\mathbf{z}}$, and thus $\hat{s} = u$. As $\max\{v_{s:u-1}, 0\} = \hat{z}_{u-1} > \hat{z}_u \geq 0$; hence, by Propositions 2 and 3, $\max\{v_{s:u-1}, 0\} = v_{s:u-1} = \hat{z}_{u-1} > \hat{z}_u = \max\{0, v_{u:\hat{t}}\} \geq v_{u:\hat{t}}$ and so $\mathcal{G}_{s:t}$ is not coherent, a contradiction.
2) $s > \hat{s}$: the proof is similar to the first case.
3) $s = \hat{s}$: together with the assumption that $t = \hat{t}$ and Proposition 3, we have $z_u = v_{s:t} = \hat{z}_u$, a contradiction. ∎

Without loss of generality, we assume that $t > \hat{t}$.

*Lemma 4:* $v_{\hat{s}:\hat{t}} > v_{\hat{t}+1:t}$.

*Proof:* Let the groups in $\hat{\mathbf{z}}$ following $\mathcal{G}_{\hat{s}:\hat{t}}$ be $\mathcal{G}_{\hat{t}+1:\hat{t}_1}, \mathcal{G}_{\hat{t}_1+1:\hat{t}_2}, \ldots, \mathcal{G}_{\hat{t}_{k-1}+1:\hat{t}_k}$, where $\hat{t}_k$ is the smallest number such that $\hat{t}_k \geq t$ (Fig. 8). Since $\hat{\mathbf{z}}$ satisfies Propositions 2 and 3, we have

$$v_{\hat{s}:\hat{t}} > v_{\hat{t}+1:\hat{t}_1} > v_{\hat{t}_1+1:\hat{t}_2} > \cdots > v_{\hat{t}_{k-1}+1:\hat{t}_k} = \hat{z}_{\hat{t}_k} \geq 0. \tag{27}$$

Recall that $\hat{t}_k \geq t$. Consider the following two cases.

1) $t < \hat{t}_k$: then $v_{\hat{t}_{k-1}+1:t} \leq v_{t+1:\hat{t}_k}$ because $\mathcal{G}_{\hat{t}_{k-1}+1:\hat{t}_k}$ is coherent. Hence, $v_{\hat{t}_{k-1}+1:t} \leq v_{\hat{t}_{k-1}+1:\hat{t}_k}$ by Lemma 2.
2) $t = \hat{t}_k$: obviously then $v_{\hat{t}_{k-1}+1:t} = v_{\hat{t}_{k-1}+1:\hat{t}_k}$.

Hence, in both cases, we can replace the last term $v_{\hat{t}_{k-1}+1:\hat{t}_k}$ in (27) by $v_{\hat{t}_{k-1}+1:t}$, and obtain

$$v_{\hat{s}:\hat{t}} > v_{\hat{t}+1:\hat{t}_1} > v_{\hat{t}_1+1:\hat{t}_2} > \cdots > v_{\hat{t}_{k-1}+1:t}. \tag{28}$$

Since $v_{\hat{t}+1:\hat{t}_1} > v_{\hat{t}_1+1:\hat{t}_2}$, we have $v_{\hat{t}+1:\hat{t}_1} > v_{\hat{t}+1:\hat{t}_2} > v_{\hat{t}_1+1:\hat{t}_2}$ from Lemma 2. Combining with (28), then $v_{\hat{t}+1:\hat{t}_2} > v_{\hat{t}_2+1:\hat{t}_3}$, and thus $v_{\hat{t}+1:\hat{t}_2} > v_{\hat{t}+1:\hat{t}_3} > v_{\hat{t}_2+1:\hat{t}_3}$. Repeating this for $\hat{t}_4, \hat{t}_5, \ldots$, we obtain $v_{\hat{t}+1:\hat{t}_1} > v_{\hat{t}+1:t}$. Result follows on substituting this back into (28). ∎

[9] The equality holds when $v_{s:t} = 0$.

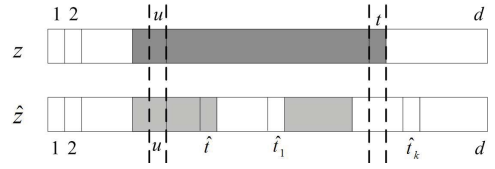



Fig. 8. Illustration for $\mathbf{z}$ and $\hat{\mathbf{z}}$ in the proof of Theorem 1.

Consider the following cases.

1) $s = \hat{s}$: then Lemma 4 implies that $\mathcal{G}_{s:t}$ is not coherent, a contradiction.
2) $s < \hat{s}$: as shown in Case 1 in the proof of Lemma 3, $\hat{s} = u$ and $\mathcal{G}_{s:\hat{s}-1}$ is a group of $\hat{\mathbf{z}}$. By Proposition 3 and Definition 1
$$v_{s:\hat{s}-1} > v_{\hat{s}:\hat{t}}. \tag{29}$$
Applying Lemma 2 to the equation in Lemma 4, we have $v_{\hat{s}:\hat{t}} > v_{\hat{s}:t}$. Combining this with (29), we obtain $v_{s:\hat{s}-1} > v_{\hat{s}:t}$, and thus group $\mathcal{G}_{s:t}$ is not coherent at $\hat{s}$, a contradiction.
3) $s > \hat{s}$: using the same argument above, $\mathcal{G}_{\hat{s}:s-1}$ is a group of $\mathbf{z}$, and $v_{\hat{s}:s-1} > v_{s:t}$. Since $\mathcal{G}_{s:t}$ is coherent, $v_{s:\hat{t}} \leq v_{\hat{t}+1:t}$. Using Lemma 2, $v_{s:t} \geq v_{s:\hat{t}}$, and so $v_{\hat{s}:s-1} > v_{s:\hat{t}}$, i.e., $\mathcal{G}_{\hat{s}:\hat{t}}$ is not coherent, a contradiction.

Hence, assumption (25) is not valid. In other words, we have proved the following.

*Proposition 7:* There is an unique $\mathbf{z}$ satisfying Propositions 2–4.

As we have shown that Propositions 2–4 hold at optimality, we only need to prove the "$\Leftarrow$" part of Theorem 1. Since Proposition 7 shows that there is only one $\mathbf{z}$ satisfying all three properties and the optimal solution of (13) clearly exists, this unique $\mathbf{z}$ must be the optimal solution.

### F. Proof of Proposition 5

Assume to the contrary that $\mathcal{G}_{s:t}$ is not coherent at some $v \in \{s, s+1, \ldots, t-1\}$ such that $v_{s:v} > v_{v+1:t}$. Consider the following three cases.

1) $v < u$: then $v_{s:v} \leq v_{v+1:u}$ as $\mathcal{G}_{s:u}$ is coherent. By Lemma 2, $v_{s:v} \leq v_{s:u} \leq v_{v+1:u}$. Using the assumption that $v_{s:u} \leq v_{u+1:t}$, we have $v_{s:v} \leq v_{u+1:t}$. By Lemma 2 again, $v_{v+1:t} \geq \min\{v_{v+1:u}, v_{u+1:t}\}$. Combining this with the previous two equations, we have $v_{s:v} \leq v_{v+1:t}$, a contradiction.
2) $u < v$: the proof is similar to Case 1.
3) $v = u$: Then $v_{s:v} > v_{v+1:t}$ reduces to $v_{s:u} > v_{u+1:t}$, which contradicts the assumption.

### G. Proof of Proposition 6

Let $\boldsymbol{\beta}^*$ be an optimal solution of
$$r^*(\boldsymbol{\gamma}) = \max_{r(\boldsymbol{\beta}) \leq 1} \boldsymbol{\gamma}^T \boldsymbol{\beta}. \tag{30}$$
Obviously, in order for (30) to be maximized, $\operatorname{sgn}(\beta_i^*) = \operatorname{sgn}(\gamma_i)$. Hence, we can assume that $\gamma_i \geq 0$ and $\beta_i^* \geq 0$. Since $\gamma_1 \geq \cdots \geq \gamma_d \geq 0$, we also have $\beta_1^* \geq \cdots \geq \beta_d^* \geq 0$. It then follows that $r(\boldsymbol{\beta}^*) = \sum_{i=1}^d w_i \beta_i^*$, where $w_i = \lambda_1 + (d-i)\lambda_2$, from the definition of the OSCAR regularizer. Moreover, the constraint in (30) should be active at optimality. Hence, (30) can be rewritten as
$$r^*(\boldsymbol{\gamma}) = \max_{\boldsymbol{\beta}} \boldsymbol{\gamma}^T \boldsymbol{\beta} \tag{31}$$
$$\text{s.t. } \beta_1 \geq \cdots \geq \beta_d \geq 0, \ \sum_{i=1}^d w_i \beta_i = 1. \tag{32}$$

*Lemma 5:* If $\beta_{s-1}^* > \beta_s^* = \cdots = \beta_u^* > \beta_{u+1}^* = \ldots \beta_t^* > \beta_{t+1}^* \geq 0$ for some $1 \leq s \leq u < t \leq d$, then $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) = (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$.

*Proof:* Assume to the contrary that $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) \neq (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$. Consider the two cases.

1) $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) > (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$. Construct $\boldsymbol{\beta}'$ from $\boldsymbol{\beta}^*$ as
$$\beta_j' = \begin{cases} \beta_u^* + \Delta \frac{\sum_{i=u+1}^t w_i}{\sum_{i=s}^u w_i}, & j = s, \ldots, u \\ \beta_{u+1}^* - \Delta, & j = u+1, \ldots, t \\ \beta_j^*, & \text{otherwise} \end{cases}$$
where $\Delta = \min\{\beta_t^* - \beta_{t+1}^*, ((\beta_{s-1}^* - \beta_s^*)\sum_{i=s}^u w_i)/(\sum_{i=u+1}^t w_i)\}$. Obviously, $\beta_{s-1}' \geq \beta_s' = \cdots = \beta_u' > \beta_{u+1}' = \beta_t' \geq \beta_{t+1}'$ and $r(\boldsymbol{\beta}') = r(\boldsymbol{\beta}^*)$ as $\sum_{i=s}^t w_i \beta_i^* = \beta_u^* \sum_{i=s}^u w_i + \beta_{u+1}^* \sum_{i=u+1}^t w_i = \beta_u' \sum_{i=s}^u w_i + \beta_{u+1}' \sum_{i=u+1}^t w_i = \sum_{i=s}^t w_i \beta_i'$. Hence, $\boldsymbol{\beta}'$ is a feasible solution of (31). Moreover, using the definition of $\boldsymbol{\beta}'$ and $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) > (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$, we have $\beta_u' \sum_{i=s}^u \gamma_i + \beta_{u+1}' \sum_{i=u+1}^t \gamma_i > \beta_u^* \sum_{i=s}^u \gamma_i + \beta_{u+1}^* \sum_{i=u+1}^t \gamma_i$, and consequently $\boldsymbol{\gamma}^T \boldsymbol{\beta}' > \boldsymbol{\gamma}^T \boldsymbol{\beta}^*$, a contradiction.
2) $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) < (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$. Construct $\boldsymbol{\beta}'$ from $\boldsymbol{\beta}^*$ as
$$\beta_j' = \begin{cases} \frac{\beta_u^* \sum_{i=s}^u w_i + \beta_{u+1}^* \sum_{i=u+1}^t w_i}{\sum_{i=s}^t w_i}, & j = s, \ldots, t \\ \beta_j^*, & \text{otherwise.} \end{cases} \tag{33}$$
Then $\beta_{s-1}' > \beta_s' = \cdots = \beta_t' > \beta_t'$ and $r(\boldsymbol{\beta}') = r(\boldsymbol{\beta}^*)$. Hence, $\boldsymbol{\beta}'$ is also feasible. Moreover, using the construction of $\boldsymbol{\beta}'$ and the condition $(\sum_{i=s}^u \gamma_i)/(\sum_{i=s}^u w_i) < (\sum_{i=u+1}^t \gamma_i)/(\sum_{i=u+1}^t w_i)$, we have $\beta_u^* \sum_{i=s}^u \gamma_i + \beta_{u+1}^* \sum_{i=u+1}^t \gamma_i < \beta_u' \sum_{i=s}^t \gamma_i$, and consequently $\boldsymbol{\gamma}^T \boldsymbol{\beta}' > \boldsymbol{\gamma}^T \boldsymbol{\beta}^*$, a contradiction.

Result follows on combining these two cases. ■

*Corollary 2:* If $\beta_u^* > \beta_{u+1}^* > 0$ for some $u$, construct $\boldsymbol{\beta}'$ as in (33). Then $\boldsymbol{\gamma}^T \boldsymbol{\beta}' = \boldsymbol{\gamma}^T \boldsymbol{\beta}^*$.

*Proof:* This is similar to that of Case 2 in Lemma 5. ■

In other words, if $\beta_u^* > \beta_{u+1}^* > 0$ for some $u$ in the optimal $\boldsymbol{\beta}^*$, one can construct another optimal $\boldsymbol{\beta}'$ such that the two groups of coefficients corresponding to $\beta_u^*$ and $\beta_{u+1}^*$ are merged. Replace $\boldsymbol{\beta}^*$ by $\boldsymbol{\beta}'$, and keep repeating this merging process. At the end, we have an optimal $\boldsymbol{\beta}^*$ such that $\beta_1^* = \cdots = \beta_j^* > 0$ and $\beta_{j+1}^* = \cdots = \beta_d^* = 0$ for some $j$.

With the optimal $\boldsymbol{\beta}^*$ constructed above, the constraint in (32) reduces to $\sum_{i=1}^j w_i \beta_i^* = \beta_1^* \sum_{i=1}^j w_i = 1$, which implies $\beta_1^* = 1/\sum_{i=1}^j w_i$. The objective in (31) then becomes

$\boldsymbol{\gamma}^T\boldsymbol{\beta}^* = \beta_1^* \sum_{i=1}^{j} \gamma_i = (\sum_{i=1}^{j} \gamma_i)/(\sum_{i=1}^{j} w_i)$. Hence, to maximize (30), we have to search for $j$ which maximizes this expression.

## REFERENCES


[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[2] L. Jiao, L. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 685–697, May 2007.

[3] K. Labusch, E. Barth, and T. Martinetz, "Simple method for high-performance digit recognition based on sparse coding," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1985–1989, Nov. 2008.

[4] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer-Verlag, 2010.

[5] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 543–550.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Stat. Soc., Ser. B*, vol. 67, no. 2, pp. 301–320, 2005.

[8] X. Shen and H. Huang, "Grouping pursuit through a regularization solution surface," *J. Amer. Stat. Assoc.*, vol. 105, no. 490, pp. 727–739, 2010.

[9] R. Jörnsten and B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL," *Bioinformatics*, vol. 19, no. 9, pp. 1100–1109, 2003.

[10] M. Dettling and P. Buhlmann, "Finding predictive gene groups from microarray data," *J. Multivariate Anal.*, vol. 90, no. 1, pp. 106–131, 2004.

[11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Stat. Soc., Ser. B*, vol. 68, no. 1, pp. 49–67, 2006.

[12] T. Hastie, R. Tibshirani, D. Botstein, and P. Brown, "Supervised harvesting of expression trees," *Genome Biol.*, vol. 2, no. 1, pp. 0003-1–0003-12, 2001.

[13] M. Park, T. Hastie, and R. Tibshirani, "Averaged gene expressions for regression," *Biostatistics*, vol. 8, no. 2, pp. 212–227, 2007.

[14] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Royal Stat. Soc., Ser. B*, vol. 67, no. 1, pp. 91–108, 2005.

[15] H. Bondell and B. Reich, "Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR," *Biometrics*, vol. 64, no. 1, pp. 115–123, 2008.

[16] S. Wu, X. Shen, and C. Geyer, "Adaptive regularization using the entire solution surface," *Biometrika*, vol. 96, no. 3, pp. 513–527, 2009.

[17] Y. She, "Sparse regression with exact clustering," *Electron. J. Stat.*, vol. 4, pp. 1055–1096, 2010.

[18] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," in *Proc. Adv. Neural Inf. Process. Syst. 24*, 2010, pp. 1558–1566.

[19] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Convex and network flow optimization for structured sparsity," *J. Mach. Learn. Res.*, vol. 12, pp. 2681–2720, Feb. 2011.

[20] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[21] L. W. Zhong and J. T. Kwok, "Efficient sparse modeling with automatic feature grouping," in *Proc. 28th Int. Conf. Mach. Learn.*, Jun. 2011, pp. 9–16.

[22] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. New York: Springer-Verlag, 2004.

[23] A. Rakotomamonjy, R. Flamary, G. Gasso, and S. Canu, "$\ell_p$-$\ell_q$ penalty for sparse linear and sparse multiple kernel multitask learning," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1307–1320, Aug. 2011.

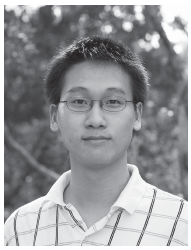
[24] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proc. 26th Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 457–464.

[25] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Convex optimization with sparsity-inducing norms," in *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. Cambridge, MA: MIT Press, 2011.

[26] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2873–2908, Dec. 2009.

[27] R. Jenatton, J. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *J. Mach. Learn. Res.*, vol. 12, pp. 2777–2824, Oct. 2011.

[28] F. Bach, "Structured sparsity-inducing norms through submodular functions," in *Proc. Adv. Neural Inf. Process. Syst. 24*, 2010, pp. 118–126.


**Leon Wenliang Zhong** received the Bachelors and Masters degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree in computer science with the Hong Kong University of Science and Technology, Kowloon, Hong Kong.

His current research interests include machine learning, convex optimization, and computational intelligence.



**James T. Kwok** received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 1996.

He was with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, as an Assistant Professor. He is currently a Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His current research interests include kernel methods, machine learning, pattern recognition, and artificial neural networks.

Dr. Kwok received the IEEE Outstanding Paper Award in 2004 and the Second Class Award in Natural Sciences from the Ministry of Education, China, in 2008. He has been a Program Co-Chair for a number of international conferences. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and *Neurocomputing*.