Balancing Statistical and Computational Precision and Applications to Penalized Linear Regression with Group Sparsity

Mahsa Taheri  
Department of Mathematics  
Ruhr-University Bochum  
44801 Bochum, Germany  
MAHSA.TAHERI@RUB.DE

Néhémé Lim  
Department of Statistics  
University of Connecticut  
Storrs, CT 06269-4120, USA  
NEHEMY.LIM@UCONN.EDU

Johannes Lederer  
Department of Mathematics  
Ruhr-University Bochum  
44801 Bochum, Germany  
JOHANNES.LEDERER@RUB.DE

Editor:

Abstract

Due to technological advances, large and high-dimensional data have become the rule rather than the exception. Methods that allow for feature selection with such data are thus highly sought after, in particular, since standard methods, such as cross-validated lasso and group-lasso, can be challenging both computationally and mathematically. In this paper, we propose a novel approach to feature selection and group feature selection in linear regression. It consists of simple optimization steps and tests, which makes it computationally more efficient than standard approaches and suitable even for very large data sets. Moreover, it satisfies sharp guarantees for estimation and feature selection in terms of oracle inequalities. We thus expect that our contribution can help to leverage the increasing volume of data in Biology, Public Health, Astronomy, Economics, and other fields.

Keywords: high-dimensional regression, group feature selection, oracle inequalities
1. Introduction

Contemporary data sets are often large and high-dimensional: they contain many parameters and samples and the number of parameters rivals or even exceeds the number of samples. On the other hand, one can often assume that the data generating model is sparse or group sparse, that is, that only a small number of parameters or a small number of groups of parameters, respectively, are actually relevant. A standard objective in the analysis of large and high-dimensional data is the selection of these relevant parameters or groups of relevant parameters in a computationally feasible and mathematically reliable way. We call these objectives feature selection and group feature selection, respectively.

A standard approach to feature selection and group feature selection is regularization. The most prominent regularized estimators in our context are the lasso (Tibshirani, 1996) and the group-lasso (Yuan and Lin, 2006). These estimators minimize the squared empirical loss plus a term proportional to the sum of the Euclidean norms of groups of coefficients. The estimators have been equipped with a range of theoretical results, see Bühlmann and van de Geer (2011, Theorem 6.4 and 8.1), for example. Also, extensions of the group-lasso have been proposed in several papers, including Simon et al. (2013), which combines overall sparsity and group sparsity, Jenatton et al. (2011), which studies group sparsity with overlap, and Micchelli et al. (2013), which makes the penalty functions more flexible. However, all these estimators prompt the same three computational and statistical challenges: (1) They involve tuning parameters that can be difficult to calibrate. For example, standard calibration schemes, such as Cross-Validation, AIC, and BIC lack finite sample guarantees and square root lasso (Belloni et al., 2011), square root group-lasso (Bunea et al., 2013), and scaled lasso (Sun and Zhang, 2012), which are alternatives to the standard lasso/group-lasso designed for alleviating the calibration problem, depend on the distribution of the noise, which is unknown in practice. (2) They are minimizers of an objective function that can be solved only approximately, and it remains unclear how to specify the corresponding tolerance: small tolerances preserve statistical precision, but large tolerances accelerate computations, especially for large data. (3) They are often equipped with thresholding algorithms for feature selection, but for group feature selection, there is currently no sound guidance for how to select the corresponding cutoffs.

In this paper, we introduce a new approach that addresses these three challenges. The main novelty of our approach is that it firmly interlaces optimization and statistics. This link is established by nesting two testing schemes: First, we use a “global” testing scheme to ensure statistical accuracy. This scheme corresponds to a tuning parameter calibration. Second, we use a “local” testing scheme to level statistical and computational precision. We show that our approach leads to computational and statistical guarantees, fast computations, and a well-motivated cutoff for thresholding.

The organization of the paper is as follows. In Section 2, we present our main proposal together with the theoretical results. In Section 3, we demonstrate the empirical performance of our proposed approach both on simulated and real data. In Section 4, we conclude with a discussion. Most of the proofs are given in the Appendix.
Notation

For any set $A$ and any vector $\beta \in \mathbb{R}^p$, we denote by $\beta_A \in \mathbb{R}^p$ the vector that equals $\beta$ on the coordinates in $A$ and is zero on the coordinates in the complement of $A$.

To disentangle theory and practice, we use bars $\overline{\quad}$ to refer to theoretical estimators (for example, one cannot compute an exact lasso solution in finite time) and tildes $\tilde{\quad}$ to practical surrogates (for example, one can approximate a lasso solution within any non-zero tolerance).

2. Methodology

In this section, we state the statistical framework and introduce our estimation approach along with sharp guarantees for it.

2.1 Statistical Setup

A common objective in statistics is to estimate an unknown target $\beta^* \in \mathcal{B}$ in some space $\mathcal{B}$ with respect to some loss function $d : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$. For example, one might want to estimate $\beta^*$ in linear regression models of the form $y = X\beta^* + u$, where $y \in \mathbb{R}^n$ is the outcome, $X \in \mathbb{R}^{n \times p}$ the design matrix, $u \in \mathbb{R}^n$ the random noise, and $\beta^* \in \mathbb{R}^p$ the unknown target, and where the loss function might be $\beta \mapsto \|\beta - \beta^*\|_q$ for a $q \in [1, \infty]$. For this, one wants to select a suitable family of estimators $\mathcal{F}_R = \{\overline{\beta}_r : r \in \mathcal{R}\}$ indexed by a set of tuning parameters $\mathcal{R}$ and then select a tuning parameter that mimics an unknown “optimal” tuning parameter $r^* \in \mathcal{R}$. For example, it is well known that $r^* = 2\|X^\top u\|_\infty$ can be a suitable tuning parameter for the lasso—see (Lederer et al., 2019) and references therein, for example. This objective includes three main questions:

1. How to select the tuning parameter in theory and practice?
2. How to compute the estimators?
3. How to choose cutoffs for feature selection?

The goal of this section is to content with these three challenges in a range of settings. We consider a general setup with a symmetric $d$ that satisfies the triangle inequality, a non-empty target space $\mathcal{B}$, and a non-empty set of tuning parameters $\mathcal{R} \subset \mathbb{R}$. In view of how estimators are implemented, we can assume without loss of generality that $\mathcal{R}$ is finite. We also consider increasing functions $f, g : \mathcal{R} \to [0, \infty)$ and a set of estimators $\tilde{\mathcal{F}}_{\mathcal{R}} := \{\tilde{\beta}_r : r \in \mathcal{R}\}$ that satisfy the following feasibility conditions:

**Definition 1 (Feasibility)** We say that the set of estimators $\mathcal{F}_R$ is $(d, f, g, \beta^*, r^*, \tilde{\mathcal{F}}_{\mathcal{R}})$-feasible if it satisfies the statistical bounds

$$d(\overline{\beta}_r, \beta^*) \leq f(r) \quad \forall r \geq r^*$$

and if the estimators can be approximated by $\tilde{\mathcal{F}}_{\mathcal{R}}$ such that it satisfies the computational bounds

$$d(\tilde{\beta}_r, \beta^*) \leq g(r) \quad \forall r \geq r^*$$.  


Since $\tilde{\beta}^r$ can be computed but usually not $\bar{\beta}^r$, practitioners eventually care about $\tilde{\beta}^r$ (with a suitable tuning parameter). The functions $d$, $f$, and $g$ need to be specified, while $\beta^*$ and $r^*$ do not need to be known. All functions and quantities, including $\beta^*$, $r^*$, and $\mathcal{R}$, may be random. We give specific examples that fit Definition 1 in the following section.

In general, Assumption (1) means that the estimators satisfy statistical bounds, such as an oracle inequality. Assumption (2) means that there are feasible algorithms to approximate the estimators to a sufficient precision. In this sense, $\tilde{\mathcal{F}}_R$ is supposed to be a set of computationally feasible estimators; one might simply set $\tilde{\mathcal{F}}_R = \mathcal{F}_R$ if $\mathcal{F}_R$ is already satisfactory from a computational perspective, but this is not the generic case. The function $f$ is typically dictated by theory, while $g$ must be chosen by the user. Traditionally, estimators in high-dimensional statistics are supposed to be computed “until convergence”, which means either that $g(r) = 0$, which is not possible in practice, or “$g(r) \approx 0$”, but it remains unclear what this actually means. A crucial feature of our approach is that it gives a clear guidance: $g(r) = f(r)$. We show mathematically and empirically that this choice balances—in a sense optimally—statistical and computational precision.

2.2 Proposed Method and Theoretical Results

We now introduce our general scheme together with its theoretical guarantees. We also specify it to group feature selection in linear regression models.

2.2.1 General Scheme

Algorithm 1 contains our main proposal. The algorithm gets the data and a set of tuning parameters $\mathcal{R}$ as inputs and returns as outputs the approximated estimator $\tilde{\beta}^\hat{r}$ and the tuning parameter $\hat{r}$. At each step, which is marked by a tuning parameter $\tilde{r}$, the algorithm approximates the estimator $\bar{\beta}^\tilde{r}$ up to a sufficient precision to satisfy computational bounds and then tests if the algorithm needs to be continued. It starts with the largest tuning parameter in $\mathcal{R}$ and then proceeds in descending order. The only computations necessary are essentially the computations of $\tilde{\beta}^{\max\{r \in \mathcal{R}\}}, \ldots, \tilde{\beta}^\tilde{r}$. The algorithm always converges (assuming the computations of the estimators converge). Following this scheme leads to a sharp theoretical guarantee:

**Theorem 2 (Guarantee for $\hat{r}$ and $\tilde{\beta}^\hat{r}$)** If $\mathcal{F}_R$ is $(d, f, g, \beta^*, r^*, \bar{\mathcal{F}}_R)$-feasible, the outputs $\hat{r}$ and $\tilde{\beta}^\hat{r}$ of Algorithm 1 satisfy

\begin{equation}
\hat{r} \leq r^*,
\end{equation}

\begin{equation}
d(\tilde{\beta}^\hat{r}, \beta^*) \leq 3f(r^*) + 3g(r^*),
\end{equation}

and

\begin{equation}
\mathbb{E}[d(\tilde{\beta}^\hat{r}, \beta^*)] \leq 3\mathbb{E}f(r^*) + 3\mathbb{E}g(r^*).
\end{equation}

Inequality (3) states that the estimated tuning parameter is bounded by the optimal tuning parameter. This result will be useful in thresholding small coefficients. Inequality (4) states that the selected estimator provides optimal performance up to the constant factor 3. The bound consists of two terms, namely $3f(r^*)$, which relates to the statistical precision.
Balancing Statistical and Computational Precision

Algorithm 1: Main Algorithm

| Inputs : data, $\mathcal{R}$ |
| Outputs: $\hat{r} \in \mathcal{R}, \hat{\beta}^\hat{r} \in \mathcal{B}$ |

1. Initialize tuning parameter: $\hat{r} \leftarrow \max\{r \in \mathcal{R}\}$;
2. Approximate corresponding estimator: compute $\hat{\beta}^\hat{r}$ such that $d(\hat{\beta}^\hat{r}, \bar{\beta}^\hat{r}) \leq g(\hat{r})$;
3. while $\hat{r} \neq \min\{r \in \mathcal{R}\}$ and

   $\exists r \in \mathcal{R} \setminus [\hat{r}, \infty)$ such that $d(\hat{\beta}^\hat{r}, \hat{\beta}^r) > f(r) + g(r)$ do

4. Update outputs: $\hat{r} \leftarrow \hat{r}; \hat{\beta} \leftarrow \hat{\beta}^\hat{r}$;
5. Go to next tuning parameter: $\hat{r} \leftarrow \max\{r \in \mathcal{R} \setminus [\hat{r}, \infty)\}$;
6. Approximate corresponding estimator: compute $\hat{\beta}^\hat{r}$ such that $d(\hat{\beta}^\hat{r}, \hat{\beta}^\hat{r}) \leq g(\hat{r})$;
7. end

and which is, up to the factor 3, the statistical bound that we have seen in Equation (1), and $3g(r^*)$, which relates to the computational precision and which, again up to the factor 3, is the computational bound that we have seen in Equation (2). Inequality (5) is the same as (4) but in expectation.

Proof [Theorem 2] We prove the three claims in order.

Claim 1: We first prove that $\hat{r} \leq r^*$. We show this by contradiction and assume that $\hat{r} > r^*$. By definition of $\hat{r}$, this means that there are tuning parameters $r', r'' \geq r^*$ such that

$$d(\hat{\beta}^{r'}, \hat{\beta}^{r''}) > f(r') + g(r') + f(r'') + g(r'').$$

On the other hand, by the assumed symmetry and triangle inequality of $d$, it holds that

$$d(\hat{\beta}^{r'}, \hat{\beta}^{r''}) \leq d(\hat{\beta}^{r'}, \beta^*) + d(\hat{\beta}^{r''}, \beta^*).$$

Using the symmetry and triangle inequality of $d$ again and combining with Assumptions (1) and (2) (statistical and computational bounds) yields for the first term

$$d(\hat{\beta}^{r'}, \beta^*) \leq d(\hat{\beta}^{r'}, \beta^*) + d(\hat{\beta}^{r''}, \beta^') \leq f(r') + g(r'),$$

and similarly for the second term

$$d(\hat{\beta}^{r''}, \beta^*) \leq f(r'') + g(r'').$$

It follows that

$$d(\hat{\beta}^{r'}, \hat{\beta}^{r''}) \leq f(r') + g(r') + f(r'') + g(r''),$$

which contradicts the initial display, and thus concludes the proof of the first inequality.

Claim 2: We now prove that

$$d(\hat{\beta}^\hat{r}, \beta^*) \leq 3f(r^*) + 3g(r^*).$$
For this, we first use the symmetry and triangle inequality of $d$ to find
\[ d(\tilde{\beta}^r, \beta^*) \leq d(\tilde{\beta}^r, \beta^r) + d(\beta^r, \beta^*). \]
The first term can be bounded similarly as in Claim 1. We find
\[ d(\tilde{\beta}^r, \beta^r) \leq f(r^*) + g(r^*). \]
The second term can be bounded by virtue of the definition of the test and $\hat{r} \leq r^*$ according to the Claim 1. We find
\[ d(\tilde{\beta}^r, \beta^*) \leq 3f(r^*) + 3g(r^*), \]
as desired.

Claim 3: We finally prove that
\[ \mathbb{E}[d(\tilde{\beta}^r, \beta^*)] \leq 3\mathbb{E}f(r^*) + 3\mathbb{E}g(r^*). \]
This follows directly from Claim 2 by taking expectations.

Theorem 2 provides optimal guarantees for one tuning parameter/estimator, namely the output of Algorithm 1. However, it can be beneficial to have some flexibility in practice. In the following, we relax the result accordingly.

Lemma 3 (Relaxed guarantee for $\hat{r}$ and $\tilde{\beta}^\hat{r}$) If $\mathcal{F}_R$ is $(d, f, g, \beta^*, r^*, \tilde{\mathcal{F}}_R)$-feasible, any estimator $\tilde{\beta}^r$, $r \geq \hat{r}$ with $\hat{r}$ from Algorithm 1, satisfies
\[ d(\tilde{\beta}^r, \beta^*) \leq 3f(\max\{r^*, r\}) + 3g(\max\{r^*, r\}). \]
This result demonstrates that one can decide to take larger tuning parameters in practice and still have some control over the loss.

We conclude with two remarks. First, $\mathcal{F}_R$ can be arbitrarily large; in particular, there are no entropy terms associated with the size of $\mathcal{F}_R$. Second, while we develop a specific pipeline for penalized linear regression below, the results in this section are completely model independent and, therefore, might be useful much more generally.

2.2.2 Application to Penalized Linear Regression

We can apply our general scheme in Algorithm 1 to the linear regression model
\[ y = X\beta^* + u, \]
where $y \in \mathbb{R}^n$ is the outcome, $X \in \mathbb{R}^{n \times p}$ the design matrix, $\beta^* \in \mathbb{R}^p = \mathcal{B}$ the regression vector with support $S \subset \{1, \ldots, p\}$, and $u \in \mathbb{R}^n$ the random noise. Our focus is the
estimation of $\beta^*$ in a measure of accuracy $d$ that is induced by a norm: $d(\beta, \beta') = |\beta - \beta'|$. For this, we consider the objective function

$$\ell(\beta, r) := \frac{1}{2} |y - X\beta|_2^2 + r||\beta||^\dagger,$$  

where $\beta \in \mathbb{R}^p, r \in \mathbb{R}$, and $||\cdot||^\dagger$ is the dual norm of $||\cdot||$. We thus propose to use estimators $F_R = \{\bar{\beta}^r : r \in \mathbb{R}\}, \bar{\beta}^r \in \text{argmin}_{\beta \in \mathbb{R}^p} \ell(\beta, r).$ (7)

Our only condition on the regularizer is decomposability with respect to $S$, that is, $||\beta||^\dagger = ||\beta_S||^\dagger + ||\beta_S^c||^\dagger$ for all $\beta \in \mathbb{R}^p$, which holds for many standard regularizers, see Negahban et al. (2012, Section 2.2), Wainwright (2014, Section 3.2), and others. A sufficient condition for satisfying the statistical bound in Definition 1 for estimators in Equation (7) is as follows:

**Lemma 4 (f in penalized linear regression)** Assume there is a constant $c > 0$ such that

$$|\delta| \leq \frac{|X^\top X\delta|}{nc} \quad \forall \delta \in \mathbb{R}^p : ||\delta_S^c||^\dagger \leq 3||\delta_S||^\dagger.$$

Then, $F_R$ satisfies

$$||\beta^* - \bar{\beta}^r|| \leq f(r) \quad \forall r \geq r^*$$

for $f(r) = 3r/(2nc)$ and $r^* = 2|X^\top u|$, where $\delta := \beta^* - \bar{\beta}^r$. The condition in the upper display of the lemma is a variant of the classical compatibility conditions (Koltchinskii, 2009; van de Geer and Bühlmann, 2009). If $||\cdot|| \equiv ||\cdot||_\infty$ for example, it coincides with the $\ell_\infty$-restricted eigenvalue condition formulated in Chichignoud et al. (2016, Equation (7)); in this sense, Lemma 4 is a generalization of their Lemma 5. We defer the proof of Lemma 4 to Appendix A. Now, plugging Lemma 4 into Theorem 2 yields the following result.

**Corollary 5 (Estimation in linear regression)** Suppose that the conditions of Lemma 4 are met for a linear regression model with objective function in Equation (6) and computational bound is satisfied by Algorithm 1 (set $g = f$), then, the outputs of Algorithm 1 satisfy

$$\hat{r} \leq r^*,$$  

$$d(\bar{\beta}^\hat{r}, \beta^*) \leq 9r^*/(nc),$$  

and

$$\mathbb{E}[d(\bar{\beta}^\hat{r}, \beta^*)] \leq \mathbb{E}[9r^*/(nc)].$$

This equips the proposed algorithm with guarantees for its estimation error in linear regression models.

Besides estimation, our approach can also be applied to feature selection. Feature selection, or more generally group feature selection consists of estimating $S := \{j \in \{1, \ldots, k\} : \beta^*_G \neq 0_{p_j}\}$, where $G^1, \ldots, G^k$ is a partition of $\{1, \ldots, p\}$ and $p_j$ the number of predictors in group $G^j$. To do this, we need a thresholding scheme as a subsequent step to the estimation.
Specifically suited base estimators for feature selection are sparsity-inducing methods, such as the group-lasso:

$$\hat{\beta}^r \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} ||y - X\beta ||^2_2 + r \sum_{j=1}^{k} \sqrt{p_j} ||\beta_{G_j} ||_2 \right\},$$

which corresponds in our framework to the loss

$$d(\beta, \beta') := \max_{j \in \{1, \ldots, k\}} ||\beta_{G_j} - \beta'_{G_j} ||_2 / \sqrt{p_j} \quad \forall \beta, \beta' \in \mathbb{R}^p. \quad (12)$$

An estimate of the support with our thresholding scheme is

$$\hat{S} := \{j \in \{1, \ldots, k\} : ||\tilde{\beta}^r_{G_j} ||_2 / \sqrt{p_j} > 9\hat{r} / (nc) \}, \quad (13)$$

where \(\hat{r}\) and \(\tilde{\beta}^r\) are the outputs of Algorithm 1 with \(f(r) = g(r) = 3r / (2nc)\) (for balancing statistical and computational precision). Assuming the above mentioned compatibility condition and an equally standard beta-min condition (Bühlmann and van de Geer, 2011)

$$\min_{j \in S} ||\beta^*_G ||_2 / \sqrt{p_j} > 18r^*/ nc, \quad (14)$$

we obtain a sharp guarantee for \(\hat{S}\).

**Lemma 6 (Group feature selection)** Under the stated conditions, it holds that

$$\hat{S} = S.$$  

According to the results of Lemma 6, Equation (13) gives us theoretically justified cutoffs for thresholding with a sharp guarantee for (group) feature selection. This guarantee takes both the tuning parameter calibration and the computational tolerances of implementations into account.

Algorithm 2 is a specification of our general scheme for group feature selection in linear regression models. Recall that bars \(\bar{\cdot}\) refer to theoretical estimators, while tildes \(\tilde{\cdot}\) refer to computable surrogates. We call the algorithm group-FOS (group-Fast and Optimal Selection) for convenient reference. The three challenges described in the introduction are now addressed as follows: (1) The stopping point on the tuning parameter path is determined via AV-tests (Chichignoud et al., 2016), which contrasts estimators in terms of the loss function (line 19 of Algorithm 2). Note that here, Equation (12) is used as the loss function. (2) We use the duality gap to ensure that the computational bound is reached. For any regression vector \(\tilde{\beta}\) and any dual feasible variable \(\tilde{\nu}\), the duality gap is \(\Delta(\tilde{\beta}, \tilde{\nu}) := \ell(\tilde{\beta}, r) - D(\tilde{\nu}, r)\), where \(D(\tilde{\nu}, r)\) is the dual function, and it is well known (Borwein and Lewis, 2010) that \(\Delta(\tilde{\beta}, \tilde{\nu})\) is an upper bound of \(\ell(\tilde{\beta}, r) - \ell(\tilde{\beta}', r)\). This upper bound ensures that the required precision is reached. Importantly, we do not need to solve the dual problem of the group-lasso, but instead, we only require a dual point, which can be found with an explicit expression. We refer to Lemma 8 (Appendix A) to show the connection between computational bound and duality gap and Appendix B regarding
the construction of feasible dual points. As the optimization algorithm, one could select proximal gradient descent (Beck and Teboulle, 2009), coordinate descent (Friedman et al., 2010), or other techniques. We opted for the first one; the corresponding updates in line 14 then read for each group

$$\tilde{\beta}_{Gj} \mapsto T_{r(\sqrt{p_j}/L)}(\tilde{\beta}_{Gj} - \frac{1}{L}X_{:,Gj}^T(X\tilde{\beta} - y)),$$

where $T$ is the block soft-thresholding operator defined by $T_b(a_{Gj}) := \max(1 - b/|a_{Gj}|_2, 0)a_{Gj}$ for $j \in \{1, \ldots, k\}$ and where $L > 0$ is the step size determined by backtracking. (3) We finally use Equation (13) for thresholding, which guarantees correct (group) feature selection (Lemma 6).

Neglecting all the intricate details, our method is simply a roughly computed group-lasso estimate with a subsequent thresholding of the elements.

As initialization, we choose the all-zeros vector in $\mathbb{R}^p$, reflecting our assumption that many groups are inactive. Since we are limited to finitely many computations in practice, we consider finite sequences $r_1 = r_{\text{max}} > r_2 > \cdots > r_M = r_{\text{min}} > 0$. The concrete choice follows the ones used in standard implementations (Friedman et al., 2010): we use a logarithmically spaced grid of size $M = 100$, set $r_{\text{max}} := \max_{j \in \{1, \ldots, k\}} |X_{:,Gj}^Ty|_2/\sqrt{p_j}$ to the smallest tuning parameter such that $\tilde{\beta}^r = 0$, and define $r_{\text{min}} := r_{\text{max}}/h$ as a fraction of $r_{\text{max}}$. Standard choices for $h$ range from 100 to 10 000. On a very high level, assuming bounded group sizes, it holds that $r_{\text{max}}/r^* \approx \max_{j \in \{1, \ldots, k\}} |X_{:,Gj}^Ty|_2/\max_{j \in \{1, \ldots, k\}} |X_{:,Gj}^Tu|_2 \approx n \sum_{j=1}^k \sqrt{p_j}|\beta_{Gj}^*|_2/\sqrt{n} \approx \sqrt{n}$. To ensure that $r_{\text{min}} < r^*$ on our data sets, we thus select $h := 1000$. Finally note that our theoretical results hold for any types of grids (also for continuous ranges of $r$) and because of the warm starts and the early stopping, the computational complexity of group-FOS depends only very mildly on $M$ and $r_{\text{min}}$. Finally, $c = 2$ and $\gamma = 1$ are considered global constants.

Computationally, the proposed method has two main advantages: First, only a part of the tuning parameter path needs to be computed, more precisely, only the part with large and moderate tuning parameters. Second, only very rough computations are required; in particular, since a large tolerance can be accepted for large tuning parameters, only very small numbers of optimization cycles (in practice, often zero to five) are required per tuning parameter.

3. Empirical performance

We demonstrate the computational efficiency and the empirical accuracy of the proposed method. To obtain a comprehensive overview, we consider a variety of experiments for feature selection and group feature selection, including synthetic data as well as biological and financial applications. group-FOS can be adapted easily to handle data without structure; we call this version FOS for convenient reference. We compare FOS to the lasso with Cross-Validation (lassoCV), which is currently the most popular method for feature selection, and with the non-convex approaches SCAD (Fan and Li, 2001) and MCP (Zhang, 2010); and we compare group-FOS with group-lasso with Cross-Validation (group-lassoCV). In Section 3.1, we consider synthetic data sets to verify the efficiency of FOS and group-FOS in regressions with moderately large data (up to 10 000 samples and parameters). In Sec-
Algorithm 2: group-FOS scheme for group feature selection in linear regression

Inputs: \( y \in \mathbb{R}^n; X \in \mathbb{R}^{n \times p}; r_1 = r_{\text{max}} > r_2 > \ldots > r_M > 0 \)

Outputs: \( \hat{\beta} \in \mathbb{R}^p; \hat{S} \subset \{1, \ldots, k\} \)

1 Initialization : \( \text{statsCont} := \text{true}; \text{statsIt} := 1; \hat{\beta}^r_1 := 0; \hat{r} := r_M; \)
2 while \( \text{statsCont} = \text{true} \) AND \( \text{statsIt} < M \) do
3     \( \text{statsIt} := \text{statsIt} + 1; \)
4     \( \text{compCont} := \text{true}; \)
5     \( \text{betaOld} := \hat{\beta}^{\text{statsIt} - 1}; \)
6     while \( \text{compCont} = \text{true} \) do
7         Compute a dual feasible point \( \tilde{\nu}^{\text{statsIt}}_r \);
8         Compute the duality gap \( \Delta(\hat{\beta}^{\text{statsIt}}_r, \tilde{\nu}^{\text{statsIt}}_r) \);
9         if \( \Delta(\hat{\beta}^{\text{statsIt}}_r, \tilde{\nu}^{\text{statsIt}}_r) \leq r^{2}_{\text{statsIt}} \gamma/n^2c^2 \) then
10            \( \hat{\beta}^{\text{statsIt}}_r := \text{betaOld}; \)
11            \( \text{compCont} := \text{false}; \)
12        else
13            for \( j = 1, \ldots, k \) do
14                \( \hat{\beta}^{\text{statsIt}}_{Gj} := T^{\text{statsIt}}_r \sqrt{p_j}/L \left( \text{betaOld}_{Gj} - X^\top \cdot \text{betaOld} - y \right)/L; \)
15            end
16            \( \text{betaOld} := \hat{\beta}^{\text{statsIt}}_r; \)
17        end
18    end
19    \( \text{statsCont} := \prod_{i=1}^{\text{statsIt}} \{ \max_{j \in \{1, \ldots, k\}} \| \hat{\beta}^{\text{statsIt}}_{Gj} - \hat{\beta}^r_{Gj} \|_2/\sqrt{p_j}(r_{\text{statsIt}} + r_i) - (3/cn) \leq 0 \}; \)
20 end
21 if \( \text{statsCont} = \text{false} \) then
22     \( \hat{r} := r_{\text{statsIt} - 1}; \)
23 end
24 \( \hat{S} := \{ j \in \{1, \ldots, k\} : \| \hat{\beta}^\hat{r}_{Gj} \|_2/\sqrt{p_j} > 9\hat{r}/(nc) \} \)

In Section 3.2, we show the scalability of FOS by analyzing a financial data set with more than 150,000 parameters. (The application to even larger regression data is currently limited by the memory restrictions in MATLAB; a future C/Fortran implementation could remove this limitation.) In Section 3.3.1, we learn a biological network with a neighborhood selection scheme. Each of the corresponding regressions comprises only 500 samples and 1000 parameters, but since 1000 such regressions are needed, the computational complexity can easily render standard methods infeasible. Finally in Section 3.3.2, we show the efficiency of group-FOS in classification on biological data.

All computations are conducted with MATLAB and are run on an Intel Core(TM) i5-3470 CPU(3.20GHz). FOS and group-FOS are implemented using the SPAMS package (Bach et al., 2011) coded in C++, which uses the duality gap as convergence criterion.
The frequency for checking the duality gap is set to every 1 iteration in the optimization function. FOS is compared with two lassoCV implementations: First, lassoCV is implemented analogously to FOS using the SPAMS package and a 10-fold Cross-Validation with warm starts. This implementation, called lassoCV_{SPAMS} in the following, is the most appropriate one for comparisons with the FOS implementation. However, much work has gone into efficient implementations of lassoCV. Therefore, we also use the well-known glmnet package (Friedman et al., 2010) and call the corresponding implementation lassoCV_{glmnet}. However, these results must be treated with reservation, because glmnet cannot be calibrated to the same convergence criterion as our implementation. More precisely, the convergence criterion in glmnet needs to be specified in terms of maximum change in the objective, which does not coincide with the criterion in our algorithm and with the convergence of the estimator itself as needed in the theory. One could also argue that comparing FOS with lassoCV_{glmnet} is not fair in any case, because glmnet exploits additional geometric properties of the lasso (such as screening rules). These additional properties could also be used in our scheme, but their implementation is deferred to future work. In any case, we demonstrate that even in its current version, FOS can outperform both lassoCV_{SPAMS} and lassoCV_{glmnet}. For SCAD and MCP, we use the SparseReg toolbox (Zhou et al., 2012). The tuning parameters that balance the fitting and penalty are set via Cross-Validation. However, SCAD and MCP each also contain a second tuning parameter that determines the shape of the penalty: for SCAD, it is set to 3.7, which minimizes the Bayes risk; for MCP, it is set to 1, the default value in the SparseReg toolbox. group-lassoCV is implemented analogously to group-FOS, using the SPAMS package and a 10-fold Cross-Validation with warm starts.

Code can be found under github.com/LedererLab.

3.1 Synthetic data

In this section, we use synthetic data to demonstrate the empirical performance of FOS and group-FOS. For FOS, data are generated from linear regression models with \( n = 500 \) and \( p = 1000 \) and with \( n = 5000 \) and \( p = 10,000 \). Each row of the design matrix \( X \in \mathbb{R}^{n \times p} \) is sampled independently from a \( p \)-dimensional normal distribution with mean 0 and covariance matrix \((1 - \rho) I_{p \times p} + \rho \mathbb{1}_{p \times p}\), where \( I_{p \times p} \) is the identity matrix, \( \mathbb{1}_{p \times p} \) the matrix of ones, and \( \rho = 0.3 \) the correlation among the features. The design matrix is then normalized such that its columns have Euclidean norm equal to \( \sqrt{n} \). The entries of the noise \( u \in \mathbb{R}^n \) are generated according to a one-dimensional standard normal distribution. The entries of \( \beta^* \) are first set to 0 except for 10 uniformly at random chosen entries that are each set to 1 or \(-1\) with equal probability; the whole vector \( \beta^* \) is then rescaled such that the signal-to-noise ratio \( \|X\beta^*\|_2^2/n \) is equal to 5.

For group-FOS, the same setup is used except for the selection of non-zero parameters: the index set is partitioned into groups of equal length, and the non-zero parameters are those in the uniformly at random chosen “active” groups. To obtain a wider range of cases, data size, group lengths, and numbers of activated groups are also modulated.

We summarize the results for feature selection and group feature selection in Table 1 and Table 2, respectively. The computational efficiency is measured in average timing in seconds; the statistical accuracy is measured in average Hamming distance, which is the sum of the
Table 1: Average run times (in seconds) and average Hamming distances for lassoCV_{SPAMS}, lassoCV_{glmnet}, SCAD, MCP, and FOS. For the larger data set, lassoCV_{SPAMS}, SCAD, and MCP timed out on our machine, which means that they took more than one hour on average.

| Method          | Timing   | Hamming distance | Timing   | Hamming distance |
|-----------------|----------|------------------|----------|------------------|
| lassoCV_{SPAMS} | 137.15 ± 9.33 | 56.00 ± 18.37 | NA       | NA               |
| lassoCV_{glmnet}| 1.31 ± 0.07  | 50.30 ± 22.77   | 84.04 ± 2.64 | 66.40 ± 34.64   |
| SCAD            | 14.20 ± 1.14 | 51.00 ± 18.01   | NA       | NA               |
| MCP             | 14.17 ± 1.01 | 61.04 ± 20.75   | NA       | NA               |
| FOS             | 0.08 ± 0.03  | 0.6 ± 1.07      | 5.49 ± 2.71 | 0.0 ± 0.0       |

Table 2: Average run times (in seconds) and average Hamming distances for group-lassoCV and group-FOS for data that vary in group length (GL) and number of activated groups (AG).

| Method          | $n = 500, p = 1000$ | $GL = 5, AG = 1$ | $GL = 5, AG = 40$ |
|-----------------|---------------------|-----------------|------------------|
| group-lassoCV   | Timing              | Hamming distance | Timing           | Hamming distance |
| group-FOS       | 0.52 ± 0.41         | 0.0 ± 0.0       | 1.24 ± 0.24      | 70.5 ± 23.5      |

| Method          | $n = 500, p = 1000$ | $GL = 10, AG = 2$ | $GL = 10, AG = 10$ |
|-----------------|---------------------|-------------------|-------------------|
| group-lassoCV   | Timing              | Hamming distance  | Timing            | Hamming distance  |
| group-FOS       | 0.5 ± 0.47          | 35.00 ± 59.11     | 0.83 ± 0.15       | 5.00 ± 12.69      |

| Method          | $n = 5000, p = 10000$ | $GL = 10, AG = 2$ | $GL = 10, AG = 10$ |
|-----------------|-----------------------|-------------------|-------------------|
| group-lassoCV   | Timing                | Hamming distance  | Timing            | Hamming distance  |
| group-FOS       | 14.77 ± 16.67         | 12.00 ± 10.32     | 131.91 ± 18.92    | 0.0 ± 0.0         |
Balancing Statistical and Computational Precision

number of false positives and the number of false negatives. If a method timed out on our machines, we put an “NA.” We observe that FOS outperforms lassoCV_{SPAMS}, lassoCV_{glmnet}, SCAD, and MCP and that group-FOS outperforms group-lassoCV both in computational efficiency and in statistical accuracy. For large data sets, lassoCV_{SPAMS}, SCAD, MCP, and group-lassoCV timed out on our machine, which means that the average time is one hour or more.

The two computational benefits of our proposed method are illustrated in Figure 1. First, we observe that even with warm starts, lassoCV_{SPAMS} requires a large number of iterations to converge. In contrast, FOS allows for early stopping, in particular, for large tuning parameters (recall that the required precision for FOS is proportional to the tuning parameter; instead, the required precision for other methods is unknown). Moreover, Cross-Validation, BIC, AIC, and similar calibration schemes are based on the entire lasso path, while only a part of the path is required for FOS. We should remark that the same result holds for group-FOS.

![Figure 1: The red, solid line depicts the number of proximal gradient steps in FOS as a function of the tuning parameter \( r \). The blue, dashed line depicts the corresponding number of proximal gradient steps in lassoCV_{SPAMS}. Shown are the numbers for one data set in the \( n = 500 \) and \( p = 1000 \) setting.](image)

3.2 Financial data

Now, we consider a large data set to demonstrate the scalability of FOS. The data (Kogan et al., 2009) comprises \( n = 16087 \) samples and \( p = 150348 \) predictors. The goal is to use financial reports of companies to predict the volatility of stock returns. The feature representation of the financial reports is based on the calculation of TF-IDF (term frequency and inverse document frequency) of unigrams. There is no ground truth available for verifying statistical accuracies, but the data is ideal for verifying the algorithm’s scalability. The computational time of lassoCV_{glmnet} is **124.56s** and of FOS **1.94s**. Instead, lassoCV_{SPAMS}, SCAD, and MCP timed out on our machine, which again means more than one hour of computation. This shows that our methodology is indeed highly scalable to large data.
3.3 Biological data

3.3.1 Lung cancer data set

FOS can also be applied to network learning problems by estimating the local neighborhood of each node via high-dimensional regressions. In this specific application, the goal is to understand the interaction network of $p = 1000$ genes in lung cancer patients from $n = 500$ expression profiles (Guyon et al., 2008). We do neighborhood selection with the “or-rule” (Meinshausen and Bühlmann, 2006) based on FOS and lassoCV and compare the estimated graphs with the available gold standard (Statnikov et al., 2015). The results are summarized in Figure 2. Note that here, the Hamming distance is the sum of the falsely included edges and the falsely omitted edges. We find that our pipeline outmatches the standard one both in speed and accuracy.

![Figure 2: Run times (in seconds) and Hamming distances (in % of the total number of possible edges) for lassoCV, glmnet, and for FOS on the lung cancer data set. The implementations lassoCV, SCAD, and MCP timed out on our machine.](image)

4. Discussion

In view of the theoretical and empirical evidence provided above, the presented algorithm is a competitive approach for feature selection and group feature selection with large and high-dimensional data. In particular, there are no comparable theoretical guarantees that provide a connection between statistical and computational accuracy, and in our simulations and real data applications, the algorithm rivals or outmatches its competitors in terms of speed and accuracy.

Acknowledgements

We thank Haim Bar for the insightful comments on a draft version of this manuscript.
3.3.2 Breast cancer data set

Here, we consider a large data set that contains gene expressions from 60 patients with estrogen-positive breast cancer (Ma et al., 2004). The patients were treated for five years and then classified into two categories (labeled as 1 or -1) according to whether the cancer recurred or not. The original data is pre-processed as follows: first, the genes with more than 50 percent missingness are removed and all other missing values are filled by mean imputation. This reduces number of genes from 22,575 to 12,071. Then, the genes are grouped using cytogenetic position data, namely the C1 set from the GSEA method (Subramanian et al., 2005). The genes that are not recorded in the C1 set are removed, which yields a total of 4,989 genes in 270 groups, with an average group size of 18.5 genes. Finally, the data is split via 10-fold Cross-Validation and group-lassoCV and group-FOS are applied. The classification is then performed by taking the signum function on the predicted values.

We report the classification errors in Figure 3. We find that group-FOS outperforms group-lassoCV in computational efficiency and accuracy.

![Figure 3: Run times (in seconds) and classification error for group-lassoCV and group-FOS on the breast cancer data set](image)

Appendix A.

In this appendix, we provide proofs of our main results and derive two properties of vectors that are close to a solution \( \tilde{\beta}^r \) of (7): in Lemma 7, we show that their error belongs to a cone; in Lemma 8, we show that they satisfy an oracle inequality.

Proof [Lemma 3] By the symmetry and triangle inequality of \( d \),

\[
d(\tilde{\beta}^r, \beta^*) \leq d(\tilde{\beta}^r, \beta^r) + d(\beta^r, \beta^*).
\]

The first term can be bounded as in the earlier proof:

\[
d(\tilde{\beta}^r, \beta^r) \leq f(r^*) + g(r^*).
\]

For the second term, we use that \( \hat{r} \leq r, r^* \) by assumption/earlier proof and the tests in Algorithm 1 to deduce that

\[
d(\beta^r, \beta^*) \leq f(r^*) + g(r^*) + f(r) + g(r).
\]

Collecting the pieces and using and that \( f, g \) are increasing yields

\[
d(\tilde{\beta}^r, \beta^*) \leq 3f(\max\{r^*, r\}) + 3g(\max\{r^*, r\}),
\]

as desired.
\textbf{Proof} [Lemma 4]

We prove the result in two steps.

\textit{Step 1:} We first prove that $\delta := \beta^* - \bar{\beta}^r$ satisfies

$$||\delta_S||^\dagger \leq 3||\delta_S||^\dagger.$$  

By definition of $\bar{\beta}^r$, we find the basic inequality

$$\frac{1}{2}||y - X\beta^*||^2_2 + r||\beta^*||^\dagger \geq \frac{1}{2}||y - X\bar{\beta}^r||^2_2 + r||\bar{\beta}^r||^\dagger.$$  

We can now rewrite $||y - X\bar{\beta}^r||^2_2/2$ as follows:

$$\frac{1}{2}||y - X\bar{\beta}^r||^2_2 = \frac{1}{2}||y - X\beta^* + X\beta^* - X\bar{\beta}^r||^2_2$$

$$= \frac{1}{2}||y - X\beta^*||^2_2 + \langle y - X\beta^*, X\beta^* - X\bar{\beta}^r \rangle + \frac{1}{2}||X\beta^* - X\bar{\beta}^r||^2_2$$

$$= \frac{1}{2}||y - X\beta^*||^2_2 + \langle X^\top (y - X\beta^*), \beta^* - \bar{\beta}^r \rangle + \frac{1}{2}||X\beta^* - X\bar{\beta}^r||^2_2.$$  

Combining the two displays yields

$$r||\beta^*||^\dagger \geq \langle X^\top (y - X\beta^*), \beta^* - \bar{\beta}^r \rangle + \frac{1}{2}||X\beta^* - X\bar{\beta}^r||^2_2 + r||\bar{\beta}^r||^\dagger.$$  

We now use that $||X\beta^* - X\bar{\beta}^r||^2_2/2 \geq 0$ and invoke the model $y = X\beta^* + u$ to find

$$r||\beta^*||^\dagger \geq \langle X^\top u, \beta^* - \bar{\beta}^r \rangle + r||\bar{\beta}^r||^\dagger.$$  

This can be rearranged to

$$r||\bar{\beta}^r||^\dagger \leq \langle X^\top u, \bar{\beta}^r - \beta^* \rangle + r||\beta^*||^\dagger.$$  

Invoking Hölder’s inequality and the assumption that $r \geq 2||X^\top u||$ provides us with

$$\langle X^\top u, \bar{\beta}^r - \beta^* \rangle \leq ||X^\top u||||\bar{\beta}^r - \beta^*||^\dagger$$

$$\leq \frac{r}{2}||\bar{\beta}^r - \beta^*||^\dagger.$$  

Together with the above inequality, we thus find

$$r||\bar{\beta}^r||^\dagger \leq \frac{r}{2}||\bar{\beta}^r - \beta^*||^\dagger + r||\beta^*||^\dagger.$$  

We then divide both sides by $r > 0$ and find

$$||\bar{\beta}^r||^\dagger \leq \frac{1}{2}||\beta^* - \bar{\beta}^r||^\dagger + ||\beta^*||^\dagger.$$  

According to the assumed decomposability of $||\cdot||^\dagger$ with respect to $S$, we can now decompose each of these vectors into their parts on $S$ and $S^C$:

$$||\bar{\beta}^r_S||^\dagger + ||\bar{\beta}^r_{S^C}||^\dagger \leq \frac{1}{2}||\beta^*_S - \beta^r_S||^\dagger + \frac{1}{2}||\bar{\beta}^r_{S^C} - \beta^*_{S^C}||^\dagger + ||\beta^*_S||^\dagger + ||\beta^*_{S^C}||^\dagger$$

$$= \frac{1}{2}||\beta^*_S - \beta^r_S||^\dagger + \frac{1}{2}||\bar{\beta}^r_{S^C}||^\dagger + ||\beta^*_S||^\dagger.$$
We can rearrange the terms and find
\[
\frac{1}{2} \| \tilde{\beta}^r_S \|^\dagger \leq \frac{1}{2} \| \tilde{\beta}^r_S - \beta^*_S \|^\dagger + \| \beta^*_S \|^\dagger - \| \tilde{\beta}^r_S \|^\dagger.
\]
Using the reverse triangle inequality, this becomes
\[
\frac{1}{2} \| \tilde{\beta}^r_S \|^\dagger \leq \frac{3}{2} \| \tilde{\beta}^r_S - \beta^*_S \|^\dagger.
\]
Finally, setting \( \delta := \beta^* - \tilde{\beta}^r \) and multiplying both sides by 2 gives the desired inequality
\[
\| \delta_S \|^\dagger \leq 3 \| \delta_S \|^\dagger.
\]

**Step 2:** We now prove that \( F_R \) satisfies
\[
\| \beta^* - \tilde{\beta}^r \| \leq \frac{3r}{2nc} \text{ for all } r \geq 2 \| X^\top u \|.
\]
For this, we first observe that the KKT conditions for the objective function at \( \tilde{\beta}^r \) are
\[-X^\top (y - X\tilde{\beta}^r) + r\bar{\kappa} = 0_p,\]
where \( \bar{\kappa} \in \partial \| \tilde{\beta}^r \|^\dagger \). Using the model, we have
\[-X^\top (X\beta^* + u - X\tilde{\beta}^r) + r\bar{\kappa} = 0_p,\]
and rearranging yields
\[X^\top X(\beta^* - \tilde{\beta}^r) = r\bar{\kappa} - X^\top u.\]
Applying the norm on both sides and using the linearity and triangle inequality for norms gives
\[
\| X^\top X(\beta^* - \tilde{\beta}^r) \| = \| r\bar{\kappa} - X^\top u \|
\leq r \| \bar{\kappa} \| + \| X^\top u \|.
\]
Using now that \( \| \bar{\kappa} \| \leq 1 \) (since \( \bar{\kappa} \in \partial \| \tilde{\beta}^r \|^\dagger \)) and \( r \geq 2 \| X^\top u \| \) (by assumption) gives
\[
| X^\top X(\beta^* - \tilde{\beta}^r) | \leq r + \frac{r}{2} = \frac{3r}{2}.
\]
Now, we recall that \( \delta := \beta^* - \tilde{\beta}^r \) satisfies \( \| \delta_S \|^\dagger \leq 3 \| \delta_S \|^\dagger \) according to Step 1. Thus, the assumption given in the lemma entails \( | \beta^* - \tilde{\beta}^r | \leq \| X^\top X(\beta^* - \tilde{\beta}^r) \|/(nc) \), so that
\[
\| \beta^* - \tilde{\beta}^r \| \leq \frac{3r}{2nc},
\]
as desired.

**Proof** [Lemma 6]
We will first prove that $\mathcal{S}^C \subset \mathcal{S}^C$. For this, consider $j \in \mathcal{S}^C$ and note that by the triangle inequality

$$\|\beta^*_j\|_2/\sqrt{p_j} = \|\beta^*_j - \tilde{\beta}^*_j + \tilde{\beta}^*_j - \tilde{\beta}^*_j\|_2/\sqrt{p_j} \leq \|\beta^*_j - \tilde{\beta}^*_j\|_2/\sqrt{p_j} + \|\tilde{\beta}^*_j\|_2/\sqrt{p_j}.$$ 

The first term can be bounded by using Lemma 4 (having assumed the compatibility condition) and Theorem 2. Result (4):

$$\|\beta^*_j - \tilde{\beta}^*_j\|_2/\sqrt{p_j} \leq \max_{j \in \{1, \ldots, k\}} |\beta^*_j - \tilde{\beta}^*_j|_2/\sqrt{p_j} \leq 3f(r^*) + 3g(r^*) = \frac{9r^*}{2nc} + \frac{9r^*}{2nc} = \frac{9r^*}{nc}.$$ 

The second term can be bounded by using $j \in \mathcal{S}^C$ and the definition of $\mathcal{S}$:

$$|\tilde{\beta}^*_j|_2/\sqrt{p_j} \leq \frac{9\hat{r}}{nc}.$$ 

Collecting the pieces and using that $\hat{r} \leq r^*$ (Theorem 2, Result (3)) gives

$$|\beta^*_j|_2/\sqrt{p_j} \leq \frac{9r^*}{nc} + \frac{9r^*}{nc} = \frac{18r^*}{nc},$$

which means by virtue of the beta-min condition that $j \in \mathcal{S}^C$. This concludes the proof of the first part.

Next, we prove that $\mathcal{S}^C \subset \mathcal{S}^C$. Using $j \in \mathcal{S}^C$, that is $\beta^*_j = 0$, we obtain

$$\|\tilde{\beta}^*_j\|_2/\sqrt{p_j} = \|\tilde{\beta}^*_j - \beta^*_j\|_2/\sqrt{p_j} \leq \frac{9r^*}{nc},$$

where the inequality is obtained similarly to the first part of the proof.

\[\Box\]

Lemma 7 Let $b \geq 0$ be a constant, $r \geq 2r^*$ a tuning parameter, and $\bar{\beta} \in \mathbb{R}^p$ any vector that satisfies $\ell(\bar{\beta}, r) \leq \ell(\bar{\beta}', r) + b$. Then $\delta := \bar{\beta}' - \bar{\beta}$ belongs to the cone

$$\mathcal{C}(\mathcal{S}) := \left\{ \nu \in \mathbb{R}^p : |\nu_{\mathcal{S}}|_1 \leq 3\|\nu_{\mathcal{S}}\|_1 + \frac{2b}{r} \right\}.$$ 

\[\text{Proof \ (Lemma 7)}\]

Since $\ell(\bar{\beta}, r) \leq \ell(\bar{\beta}', r) + b \leq \ell(\bar{\beta}'', r) + b$, we find the basic inequality

$$\frac{1}{2}\|y - X\bar{\beta}\|_2^2 + r|\bar{\beta}|_1 \leq \frac{1}{2}\|y - X\bar{\beta}'\|_2^2 + r|\bar{\beta}'|_1 + b.$$ 

We can now rewrite $|y - X\bar{\beta}|_2^2/2$ as follows:

$$\frac{1}{2}|y - X\bar{\beta}|_2^2 = \frac{1}{2}|y - X\bar{\beta}'/2 + X\bar{\beta}'/2 - X\bar{\beta}|_2^2$$

$$= \frac{1}{2}|y - X\bar{\beta}'/2|_2^2 + \langle y - X\bar{\beta}'/2, X\bar{\beta}'/2 - X\bar{\beta} \rangle + \frac{1}{2}\|X\bar{\beta}'/2 - X\bar{\beta}\|_2^2$$

$$= \frac{1}{2}|y - X\bar{\beta}'/2|_2^2 + \langle y^\top(y - X\bar{\beta}'/2), \bar{\beta}'/2 - \bar{\beta} \rangle + \frac{1}{2}\|X\bar{\beta}'/2 - X\bar{\beta}\|_2^2.$$
Combining the two displays yields

\[
\langle X^\top (y - X\bar{\beta}^{r/2}), \beta - \bar{\beta} \rangle + \frac{1}{2} \|X\bar{\beta}^{r/2} - X\bar{\beta}\|_2 + r\|\bar{\beta}\|^\dagger \leq r\|\bar{\beta}^{r/2}\|^\dagger + b,
\]

and, by rearranging and using that \(\|X\bar{\beta}^{r/2} - X\bar{\beta}\|_2^2 / 2 \geq 0\),

\[
r\|\bar{\beta}\|^\dagger \leq \langle X^\top (y - X\bar{\beta}^{r/2}), \beta - \bar{\beta}^{r/2} \rangle + r\|\bar{\beta}^{r/2}\|^\dagger + b.
\]

Invoking Hölder’s inequality and the KKT conditions for \(\bar{\beta}^{r/2}\) provides us with

\[
\langle X^\top (y - X\bar{\beta}^{r/2}), \beta - \bar{\beta}^{r/2} \rangle \leq \|X^\top (y - X\bar{\beta}^{r/2})\|\|\bar{\beta} - \bar{\beta}^{r/2}\|^\dagger \\
\leq \frac{r}{2}\|\bar{\beta} - \bar{\beta}^{r/2}\|^\dagger.
\]

Combining the two displays yields

\[
2\|\bar{\beta}\|^\dagger \leq \|\bar{\beta} - \bar{\beta}^{r/2}\|^\dagger + 2\|\bar{\beta}^{r/2}\|^\dagger + \frac{2b}{r}.
\]

Hence,

\[
2\|\bar{\beta}_S\|^\dagger + 2\|\bar{\beta}_{S^c}\|^\dagger \leq \|\bar{\beta}_S - \bar{\beta}_S^{r/2}\|^\dagger + \|\bar{\beta}_{S^c} - \bar{\beta}_{S^c}^{r/2}\|^\dagger + 2\|\bar{\beta}_S^{r/2}\|^\dagger + 2\|\bar{\beta}_{S^c}^{r/2}\|^\dagger + \frac{2b}{r} \\
= \|\bar{\beta}_S - \bar{\beta}_S^{r/2}\|^\dagger + \|\bar{\beta}_{S^c}\|^\dagger + 2\|\bar{\beta}_S^{r/2}\|^\dagger + \frac{2b}{r},
\]

where \(\bar{S} := \text{supp}(\bar{\beta}^{r/2})\). This is equivalent to

\[
\|\bar{\beta}_{S^c}\|^\dagger \leq \|\bar{\beta}_S - \bar{\beta}_S^{r/2}\|^\dagger + 2\|\bar{\beta}_S^{r/2}\|^\dagger - 2\|\bar{\beta}_S\|^\dagger + \frac{2b}{r},
\]

so that with the reverse triangle inequality

\[
\|\bar{\beta}_{S^c}\|^\dagger \leq 3\|\bar{\beta}_S - \bar{\beta}_S^{r/2}\|^\dagger + \frac{2b}{r}.
\]

Finally, setting \(\delta := \bar{\beta} - \bar{\beta}\), we get

\[
\|\delta_{S^c}\|^\dagger \leq 3\|\delta_S\|^\dagger + \frac{2b}{r}.
\]

Now, since \(r/2 \geq r^*\), Lemma 4 entails \(\|\delta_S\|^\dagger \leq \|\delta_S\|^\dagger\) and \(\|\delta_{S^c}\|^\dagger \leq \|\delta_{S^c}\|^\dagger\). Combining these two inequalities with the above display yields

\[
\|\delta_{S^c}\|^\dagger \leq 3\|\delta_S\|^\dagger + \frac{2b}{r} \leq 3\|\delta_S\|^\dagger + \frac{2b}{r},
\]

as desired.
Lemma 8 (g in penalized linear regression) For given $b \geq 0$, assume there is a constant $\gamma > 0$ such that

$$\|X\delta\|_2^2 \geq \gamma \|\delta\|_2^2$$

for all $\delta \in \mathbb{R}^p$ that satisfy $\|\delta_S\|_1 \leq 3\|\delta_S\|_1 + 2b/r$. Then, $\tilde{F}_R$ satisfies

$$\|\tilde{\beta}^* - \bar{\beta}\| \leq g(r) \quad \forall r \geq 2r^*$$

for $g(r) = \sqrt{2b/\gamma}$ and any $\bar{\beta} \in \mathbb{R}^p$ with $\ell(\bar{\beta}, r) \leq \ell(\tilde{\beta}^*, r) + b$.

Proof [Lemma 8] Since $\ell(\tilde{\beta}^*, r) \leq \ell(\bar{\beta}, r) \leq \ell(\tilde{\beta}^*, r) + b$, we find the basic equality

$$\frac{1}{2}\|y - X\tilde{\beta} + X\tilde{\beta}^* - X\bar{\beta}\|_2^2 + r\|\tilde{\beta}\|^\dagger = \frac{1}{2}\|y - X\tilde{\beta}^*\|_2^2 + r\|\tilde{\beta}^*\|^\dagger + b',$$

for some $b' \in [0, b]$. This equation is equivalent to

$$\frac{1}{2}\|y - X\tilde{\beta}^* + X\tilde{\beta}^* - X\bar{\beta}\|_2^2 + r\|\tilde{\beta}\|^\dagger = \frac{1}{2}\|y - X\bar{\beta}\|_2^2 + r\|\bar{\beta}\|^\dagger + b',$$

and hence to

$$\frac{1}{2}\|y - X\tilde{\beta}^*\|_2^2 + \langle y - X\tilde{\beta}^*, X\tilde{\beta}^* - X\bar{\beta} \rangle + \frac{1}{2}\|X\tilde{\beta}^* - X\bar{\beta}\|_2^2 + r\|\tilde{\beta}\|^\dagger = \frac{1}{2}\|y - X\tilde{\beta}^*\|_2^2 + r\|\tilde{\beta}^*\|^\dagger + b',$$

and finally

$$\frac{1}{2}\|y - X\tilde{\beta}^*\|_2^2 + \|X^\top(y - X\tilde{\beta}^*), \tilde{\beta} - \bar{\beta} \| + \frac{1}{2}\|X\tilde{\beta}^* - X\bar{\beta}\|_2^2 + r\|\tilde{\beta}\|^\dagger = \frac{1}{2}\|y - X\tilde{\beta}^*\|_2^2 + r\|\tilde{\beta}^*\|^\dagger + b'.$$

Rearranging yields

$$\frac{1}{2}\|X\tilde{\beta}^* - X\bar{\beta}\|_2^2 = \langle X^\top(y - X\tilde{\beta}^*), \tilde{\beta} - \bar{\beta} \rangle - r\|\tilde{\beta}\|^\dagger + r\|\tilde{\beta}^*\|^\dagger + b'.$$

We now recall that the KKT conditions for the objective function $f$ read

$$-X^\top(y - X\tilde{\beta}^*) + r\tilde{\kappa} = 0$$

for any vector $\tilde{\kappa}$ that satisfies $|\tilde{\kappa}| \leq 1$ and $\kappa^\top \tilde{\beta}^* = \|\tilde{\beta}^*\|^\dagger$. Plugging this into the above display yields

$$\frac{1}{2}\|X\tilde{\beta}^* - X\bar{\beta}\|_2^2 = \langle r\tilde{\kappa}, \tilde{\beta} - \bar{\beta} \rangle - r\|\tilde{\beta}\|^\dagger + r\|\tilde{\beta}^*\|^\dagger + b'$$

$$= r(\kappa^\top \bar{\beta} - \|\bar{\beta}\|^\dagger) + b',$$

and by Hölder’s inequality, $\kappa^\top \bar{\beta} \leq \|\kappa\|\|\bar{\beta}\|^\dagger \leq \|\bar{\beta}\|^\dagger$. Therefore,

$$\frac{1}{2}\|X\tilde{\beta}^* - X\bar{\beta}\|_2^2 \leq b' \leq b.$$

We finally get, setting $\delta := \tilde{\beta}^* - \bar{\beta}$,

$$\frac{1}{2}\|X\delta\|_2^2 \leq b.$$
Moreover, Lemma 7 guarantees that $\delta \in C(S)$ and thus allows us to apply the condition in the upper display of the lemma. So we also find

$$
|X\delta|_2^2 \geq \gamma \|\delta\|_2^2.
$$

Combining these two inequalities gives us

$$
\|\delta\|_2^2 \leq \frac{2b}{\gamma},
$$

which implies

$$
\|\delta\| \leq \sqrt{\frac{2b}{\gamma}},
$$

as desired.

Appendix B. Underlying optimization results

Our approach estimates the computational accuracies of iterative optimization steps. For this, the concept of duality gap guarantees the convergence of the algorithm up to the desired precision by providing an upper bound of $\ell(\hat{\beta}, r) - \ell(\bar{\beta}^r, r)$. The computation of the duality gap requires the construction of a dual point for the approximated estimator $\hat{\beta}$. In this section, we show how to find an explicit expression of a feasible dual point.

First, we give a dual formulation of the primal objective (7) (Borwein and Lewis, 2010; Ndiaye et al., 2016)

$$
\begin{align*}
\bar{\nu}^r := & \arg\max_{\nu \in \mathbb{R}^n} D(\nu, r) := -\frac{r^2}{2} |\nu - \frac{y}{r}|_2^2 + \frac{1}{2} |y|_2^2 \\
\text{subject to} & \quad \|X^\top \nu\| \leq 1
\end{align*}
$$

(16)

A primal solution $\bar{\beta}^r$ of (7) and the dual solution $\bar{\nu}^r$ of (16) are linked by the relation

$$
X\bar{\beta}^r = y - r\bar{\nu}^r.
$$

This suggests to choose as dual feasible point a rescaled version of the residuals. Given a current primal estimate $\hat{\beta}$, this yields $\bar{\nu} = s(y - X\hat{\beta})$, where $s$ is given by (Ghaoui et al., 2010)

$$
s = \min \left\{ \max \left\{ \frac{-1}{\|X^\top(y - X\hat{\beta})\|}, \frac{y^\top(y - X\hat{\beta})}{\|y - X\hat{\beta}\|_2^2} \right\}, \frac{1}{\|X^\top(y - X\hat{\beta})\|} \right\}.
$$

This choice for the coefficient $s$ ensures that $\bar{\nu}$ is the closest (in $\ell_2$-norm) point to $y/r$ in the feasible set $\{\nu \in \mathbb{R}^n : \|X^\top \nu\| \leq 1\}.$
References

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

J. Borwein and A. Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer-Verlag New York, 2010.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data: Methods, theory and applications*. Springer Ser. Statist. 2011. doi: 10.1007/978-3-642-20192-9. URL http://dx.doi.org/10.1007/978-3-642-20192-9.

F. Bunea, J. Lederer, and Y. She. The group square-root lasso: Theoretical properties and fast algorithms. *IEEE Trans. Inform. Theory*, 60(2):1313–1325, 2013.

M. Chichignoud, J. Lederer, and M. Wainwright. A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *J. Mach. Learn. Res.*, 17(1):8162–8188, 2016.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360, 2001.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. of Stat. Softw.*, 33(1):1–22, 2010. URL http://www.jstatsoft.org/v33/i01/

L. Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv:1009.4219*, 2010.

I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A. Statnikov. Design and analysis of the causation and prediction challenge. In *WCCI Causation and Prediction Challenge*, pages 1–33, 2008.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.*, 12:2777–2824, 2011.

S. Kogan, D. Levin, B. Routledge, J. Sagi, and N. Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280, 2009.

V. Koltchinskii. Sparsity in penalized empirical risk minimization. *Ann. Inst. Henri Poincaré Probab. Stat.*, 45(1):7–57, 2009. ISSN 0246-0203. doi: 10.1214/07-AIHP146. URL http://dx.doi.org/10.1214/07-AIHP146.

22
J. Lederer, L. Yu, and I. Gaynanova. Oracle inequalities for high-dimensional prediction. *Bernoulli*, 25(2):1225–1255, 2019.

X.-J. Ma, Z. Wang, P. Ryan, S. Isakoff, A. Barmettler, A. Fuller, B. Muir, G. Mohapatra, R. Salunga, J. Tuggle, and Y. Tran. A two-gene expression ratio predicts clinical outcome in breast cancer patients treated with tamoxifen. *Cancer cell*, 5(6):607–616, 2004.

N. Meinshausen and P. B"{u}hlmann. High-dimensional graphs and variable selection with the lasso. *Ann. Statist.*, 34(3):1436–1462, 2006.

C. Micchelli, J. Morales, and M. Pontil. Regularizers for structured sparsity. *Adv. Comput. Math.*, 38(3):455–489, 2013.

E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparse-group lasso. In *Advances in Neural Information Processing Systems 29*, pages 388–396, 2016.

S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statist. Sci.*, 27(4):538–557, 2012.

N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *J. Comput. Graph. Statist.*, 22(2):231–245, 2013.

A. Statnikov, S. Ma, M. Henaff, N. Lytkin, E. Efstathiadis, E. Peskin, and C. Aliferis. Ultra-scalable and efficient methods for hybrid observational and experimental local causal pathway discovery. *J. Mach. Learn. Res.*, 16(1):3219–3267, 2015.

A. Subramanian, P. Tamayo, V. Mootha, S. Mukherjee, B. Ebert, M. Gillette, A. Paulovich, S. Pomeroy, T. Golub, E. Lander, and J. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Nat. Acad. Sci. India Sect. A*, 102(43):15545–15550, 2005.

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996.

S. van de Geer and P. B"{u}hlmann. On the conditions used to prove oracle results for the Lasso. *Electron. J. Stat.*, 3:1360–1392, 2009. URL http://dx.doi.org/10.1214/09-EJS506.

M. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annu. Rev. Stat. Appl.*, 1:233–253, 2014.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B.*, 68(1):49–67, 2006.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942, 2010.
H. Zhou, A. Armagan, and D. Dunson. Path following and empirical bayes model selection for sparse regression. *arXiv:1201.3528*, 2012.