Statistical Science
2017, Vol. 32, No. 3, 367–384
DOI: 10.1214/16-STS602

Logistic Regression: From Art to Science

Dimitris Bertsimas and Angela King

Abstract. A high quality logistic regression model contains various desirable properties: predictive power, interpretability, significance, robustness to error in data and sparsity, among others. To achieve these competing goals, modelers incorporate these properties iteratively as they hone in on a final model. In the period 1991–2015, algorithmic advances in Mixed-Integer Linear Optimization (MILO) coupled with hardware improvements have resulted in an astonishing 450 billion factor speedup in solving MILO problems. Motivated by this speedup, we propose modeling logistic regression problems algorithmically with a mixed integer nonlinear optimization (MINLO) approach in order to explicitly incorporate these properties in a joint, rather than sequential, fashion. The resulting MINLO is flexible and can be adjusted based on the needs of the modeler. Using both real and synthetic data, we demonstrate that the overall approach is generally applicable and provides high quality solutions in realistic timelines as well as a guarantee of suboptimality. When the MINLO is infeasible, we obtain a guarantee that imposing distinct statistical properties is simply not feasible.

Key words and phrases: Logistic regression, computational statistics, mixed integer nonlinear optimization.

1. INTRODUCTION

Logistic regression is a common classification method when the response variable is binary. Given a response vector $\mathbf{y}_{n\times 1}$, a model matrix $\mathbf{X} = [\mathbf{X}_1', \ldots, \mathbf{X}_n'] \in \mathbb{R}^{n\times p}$, and regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$, the logistic regression model assumes $\log(P(y_i = 1 \mid \mathbf{x}_i)/P(y_i = 0 \mid \mathbf{x}_i)) = \boldsymbol{\beta}'\mathbf{x}_i$. Logistic regression minimizes the negative log-likelihood of the data

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}), \tag{1}$$

where $f(\boldsymbol{\beta}) = \sum_{i=1}^{n} -y_i(\boldsymbol{\beta}'\mathbf{x}_i) + \log(1 + \exp(\boldsymbol{\beta}'\mathbf{x}_i))$.

The logistic function was originally invented in the nineteenth century to model population growth. In the early twentieth century, it gained support as a tool for bioassay, and over the course of the twentieth century its applications grew to span many fields (see [16] for a thorough overview). The simplicity and effectiveness of the logistic regression model have made it an essential part of every statistician's toolkit today. The careful modeler often spends substantial time and effort building a high quality logistic regression model from the raw data. It is rare for the modeler to build a single model. Rather, to produce an interpretable model that successfully decouples signal from noise, the modeler usually embarks upon an iterative process of model selection and refinement. Throughout this process, she must keep in mind a set of properties that a high-quality logistic regression model will exhibit: among others, it should be parsimonious but generalizable, free of excessive multicollinearity, not overly determined by individual outliers, and of course, must cohere with the application at hand. Traditionally, balancing these competing goals to create a successful, high-quality logistic regression model has been more of an art than a science. In this paper, we propose an algorithmic approach to jointly satisfying such objectives based on optimization. We will address the special case of best subset logistic regression extending [8] that focuses on best subset linear regression, and show that we can use our approach to find subsets of variables when the number of variables is over 30,000.

Dimitris Bertsimas is Professor, Operations Research Center and Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA (e-mail: dbertsim@mit.edu). Angela King is Data Scientist, End to End Analytics, San Francisco, California, USA (e-mail: aking10@mit.edu).

1.1 The Aspirations of the Work

In regression modeling, the modeler accounts for desirable characteristics one at a time through a course of model experimentation and refitting. The final model produced may contain the desired properties, but there is no guarantee of this. Moreover, there is no guarantee that the final model is indeed the best possible model for satisfying the modeler's original goals. The intention of this work is to lay out an algorithmic framework for building high quality logistic regression models based on optimization which account for all of the modeler's original goals simultaneously. The core of this framework is a mixed integer nonlinear optimization (MINLO) problem, and we develop a method to solve this MINLO in practical time frames. If it is not possible to jointly achieve the modeler's goals, our algorithm gives a guarantee that this is indeed infeasible. If it is possible, the algorithm outputs a set of high quality models incorporating the desired properties.

1.2 Current Practice

The challenge of building a high quality logistic regression model lies in the fact that the modeler must artfully manage various competing objectives. Different modelers may approach the same data with the same objectives, but because of decisions they make along the model-building process, may wind up with very different final models.

We consulted several logistic regression textbooks ([32, 45, 30] and [19]) to better understand how modelers formally learn how to build logistic regression models. This textbook review indicated that although authors try to make readers aware of the various competing objectives, they rarely give direction on how to consider these objectives as a whole when constructing the best possible model. To wit, [45] discusses selection criteria, stepwise approaches, and testing for multicollinearity and nonlinear transformations. However, [45] notes that "Using logistic regression diagnostics... is more art than science."

Hosmer and Lemeshow [32] outlines a methodology for fitting logistic regression models that they call "purposeful selection." In our experience, many modelers follow this iterative approach in practice. For completeness, we include a summary of purposeful selection in the supplemental file [7]. For datasets for which purposeful selection can be done manually, this is a laborious but necessary task of building and rebuilding the logistic regression model. With our computational approach, we aim to eliminate this tedious step and produce a set of high quality models. Moreover, our computational approach extends to the high dimensional regime and is not limited to the case where there is a low number of potential covariates.

1.3 Contribution and Structure of the Paper

In this paper, we propose a mixed integer nonlinear optimization (MINLO) approach to model a variety of desired properties in statistical models. In Table 1, we summarize the properties we model and how they are built into the MINLO model in Section 2. Our approach provides the only methodology we are aware of to construct models that impose statistical properties in logistic regression models simultaneously. The MINLO problem is challenging to solve from an optimization perspective and we propose a tailored methodology to solve it based on modern optimization techniques which is faster than existing MINLO software. We combine outer approximation techniques in mixed integer nonlinear optimization with dynamic constraint generation, a feature of modern optimization solvers. To the best of our knowledge, we are the first to integrate the optimization-based technique of dynamic constraint generation, which automatically generates

TABLE 1
Desirable properties of a logistic regression model and how they are built into the model

Property	Paper section	MINLO model
General sparsity	2.1	Constraint (7d)
Group sparsity	2.2	Constraint (7e)
Limited pairwise multicollinearity	2.2	Constraint (7f)
Nonlinear transformations	2.2	Constraint (7g)
Robustness	2.3	Objective (7b)
Modeler expertise	2.4	Constraint (7h)
Statistical significance	2.5	Constraint (7i)
Low global multicollinearity	2.6	Constraint (7i)

and add constraints to the MINLO at certain points in the solving process, into statistical modeling. This allows us to take full advantage of the speedups in mixed integer linear optimization (MILO). We also consider the well-studied special case where the only property we would like to impose is sparsity: this is known as the best subset problem in logistic regression. In this case, we can incorporate a discrete extension of first-order methods in continuous optimization into our tailored method for solving the MINLO.

Using both real and synthetic data, we demonstrate that the overall approach is generally applicable, tractable in the sense of providing solutions in realistic timelines, and provides a guarantee of suboptimality as it is based on a MINLO model. Specifically, when the MINLO is infeasible we obtain a guarantee that imposing distinct statistical properties is simply not feasible.

The paper is structured as follows. We begin in Section 2, with a discussion of the desirable statistical properties we want the regression model to have. In Section 3, we explain the algorithmic framework which achieves these properties, including the formulation of the MINLO model. In Section 4, we give a brief review of MINLO and explain our tailored method for solving the MINLO model. In Section 5, we provide evidence of our algorithm's abilities using a wide variety of real and synthetic datasets. We conclude in Section 6.

2. DESIRABLE PROPERTIES

We outline desirable characteristics of a logistic regression model, and compare our MINLO approach to achieving these properties in logistic regression models with existing approaches in the literature.

2.1 General Sparsity

When the number of potential features is large, we often wish to identify a critical subset of features which are primarily responsible for producing the response. This leads to more interpretable models, and aids prediction accuracy by eliminating unnecessary variables to increase the model's ability to generalize.

Statistically incorporating sparsity into regression models has received a great deal of attention in the context of the *best subset problem*, which is the problem of determining the best k-feature fit in a regression model:

$$\min_{\boldsymbol{\beta}} \quad f(\boldsymbol{\beta}) \quad \text{subject to } \|\boldsymbol{\beta}\|_0 \le k, \tag{2}$$

where the ℓ_0 (pseudo)norm of a vector $\boldsymbol{\beta}$ counts the number of nonzeros in $\boldsymbol{\beta}$ and is given by $\|\boldsymbol{\beta}\|_0 = \sum_{i=1}^{p} 1(\beta_i \neq 0)$.

Furnival and Wilson [28] proposed solving Problem (2) via an implicit enumeration algorithm when $f(\boldsymbol{\beta})$ is the linear regression objective function. Hosmer et al. [31] showed that software implementing the algorithm of [28] can be used directly in the case of logistic regression as well. However, the algorithm of [28] does not scale past $p = 30$, leading much of the statistics community to view solving Problem (2) as generally intractable.

The familiar Lasso ℓ_1-penalty approach,

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} -y_i(\boldsymbol{\beta}'\mathbf{x}_i) + \log(1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)) + \lambda\|\boldsymbol{\beta}\|_1, \tag{3}$$

has been proposed in the literature. Friedman et al. [27] and [36, 37, 39] suggested various methods to solve (3). Alternative approaches based on a prior to encourage sparsity have been proposed in [24, 52] and [38].

Satoa [48] is the only work that solves a penalized version of (2) via mixed integer optimization (MIO). They approximate the log likelihood with a piecewise linear function and solve the resulting linear MIO with standard solvers. They suggest solving the full MINLO rather than a linear approximation as a direction of future research. In this paper, we will directly consider the MINLO.

2.2 Selective Sparsity

We use the term "selective sparsity" to refer to settings where we would like to constrain the joint inclusion of subsets of independent variables: group sparsity, pairwise multicollinearity and nonlinear transformations.

Group sparsity. Some applications exhibit a block or group-sparse structure, with groups of independent variables whose coefficients are either all zero or all nonzero. Categorical variables, when expressed as a collection of dummy variables, form a natural group structure. Clear group formations also appear in compressed sensing [23], microarray analysis [42] and other applications. Group sparsity has been highly studied in recent years (e.g., see [1, 53, 54]). Group Lasso, first proposed for linear regression in [53], has analogously been proposed for logistic regression [35, 44, 50].

Limited pairwise multicollinearity. A near-linear relationship between independent variables obfuscates the true contribution of each feature to the response and

leads to unstable parameter estimates. To avoid these issues and produce interpretable models, [51] recommends that "independent variables with a pairwise correlation more than 0.70 should not be included in multiple regression analysis." This can be modeled as selective sparsity; see Constraint (7f).

Detecting appropriate nonlinear transformations. Nonlinear transformations of independent variables may be able to explain the variance in the dependent variable much better than the original measured variable could. Such transformations are detected through graphical examination and trial and error, or automatically by the Box–Tidwell procedure ([12] for linear regression and [32, 45] for logistic regression).

2.3 Robustness

Robustness in logistic regression has mainly focused on developing alternative objectives to maximum likelihood which are robust against outliers. Carroll and Pederson [14] suggested using a weighted maximum likelihood estimator. [46, 10] and [17] considered robust M-estimates. See [43] for an overview.

Robust optimization directly addresses errors in the data by considering uncertainty sets for the data and calculates solutions that are immune to worst-case uncertainty under these sets (see [3] and [5]). For the logistic regression problem with data $(\mathbf{y}, \mathbf{X})$, the data associated with the independent variables have error $\mathbf{\Delta X}$ that belong to a given uncertainty set U. For example,

$$U = \{\mathbf{\Delta X} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta x}_i\|_\alpha \leq \Gamma\},$$

where $\|\mathbf{x}\|_\alpha = (\sum_{l=1}^n x_l^\alpha)^{1/\alpha}$. The robust logistic regression problem is then

$$\min_{\boldsymbol{\beta}} \max_{\mathbf{\Delta X}\in U} \sum_{i=1}^n -y_i(\boldsymbol{\beta}'(\mathbf{x}_i + \Delta\mathbf{x}_i)) + \log(1+\exp(\boldsymbol{\beta}'(\mathbf{x}_i+\Delta\mathbf{x}_i))). \tag{4}$$

The key result is as follows.

THEOREM 2.1 ([6]). *Problem* (4) *is equivalent to*

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n -y_i(\boldsymbol{\beta}'\mathbf{x}_i + (-1)^{y_i}\Gamma\|\boldsymbol{\beta}\|_{\frac{\alpha}{\alpha-1}}) + \log(1+\exp(\boldsymbol{\beta}'\mathbf{x}_i + (-1)^{y_i}\Gamma\|\boldsymbol{\beta}\|_{\frac{\alpha}{\alpha-1}})). \tag{5}$$

Note that this differs from the recent work of [49] which considers robustness in logistic regression in a distributional sense.

2.4 Modeler Expertise

There may be cases where the modeler has domain knowledge about the features in the model. In that case, she might wish to specify that certain independent variables must be included in the final logistic regression model, due to a known correlation with the response. This can be incorporated directly into the model building process by adding Constraint (7h) to Problem (7).

2.5 Statistical Significance

Statistical significance of logistic regression models is typically estimated via a likelihood ratio test, Wald's test, or a Lagrange multiplier test (also known as score test) [32]. These three tests are asymptotically equivalent, but since we consider a robustified and constrained version of maximum likelihood, none of these tests is directly applicable to our case. We will maintain an assumption-free approach by using bootstrapping methods, introduced in [22], in order to estimate confidence intervals for each feature in the model selected by our algorithm.

2.6 Low Global Multicollinearity

Ryan in [47] reports an example with all pairwise correlations ≤ 0.57 but a perfect linear relationship. Global multicollinearity can be measured by checking the condition number of the correlation matrix resulting from the submatrix of included variables. A condition number greater than 15 is usually taken as evidence of multicollinearity and a condition number greater than 30 is usually an instance of severe multicollinearity [15].

3. THE OPTIMIZATION FRAMEWORK

In this section, we describe our three-stage iterative procedure for producing high quality regression models. The stages are: (1) preprocessing, (2) building and solving the MINLO model and (3) generating any additional constraints and repeating Stage 2.

3.1 Stage 1: Preprocessing

The dataset is split randomly 50%/25%/25% into a training, validation and test set. Each set is standardized so that the training set has columns with zero mean and unit ℓ_2-norm. The modeler may also choose to set the number of robustification parameters Γ to be tested in the model (the default is 5), and ρ, the maximum pairwise correlation that will be allowed between included variables (the default is 0.7). The algorithm then generates the correlation matrix for the training data

and identifies variables which are correlated in absolute value beyond ρ, and calls this set of pairs of variables $\mathcal{HC}$, for highly correlated variables. The modeler specifies any variables which are categorical, and the algorithm expresses these as groups of dummy variables. At this point, the modeler can also specify any additional group-sparsity structure. We denote the mth set of group-sparse variables as $\mathcal{GS}_m$. The modeler can specify a set of variables to be considered for a nonlinear transformation, and the algorithm generates transformed versions of those variables. The default transformations for variable x are $x^2, x^{1/2}$ and $\log x$. We denote the mth set of transformed variables by $\mathcal{T}_m$. Finally, if the modeler has subject expertise, she can specify a set $\mathcal{I}$ of variables that must be included in the model. Then the algorithm calculates $k_{\max}$, the maximum possible subset size such that the selective sparsity and modeler expertise constraints are still feasible. We construct a graph containing vertices corresponding to each of the p potential variables and an edge between nodes i, j such that $(i, j) \in \mathcal{HC}$. Then a maximum independent set for this graph is a set such that no two vertices are adjacent. The cardinality of this set is exactly equal to $k_{\max}$, and is the objective value of the following MIO problem:

$$\begin{aligned} \text{OBJ} = \max_{\mathbf{z}} & \sum_{i=1}^{p} z_i \\ \text{s.t.} \quad & z_i + z_j \le 1 \quad \forall (i,j) \in \mathcal{HC}, \\ & z_i \in \{0,1\}, \quad i = 1, \ldots, p. \end{aligned} \tag{6}$$

Since the graph contains at least one node, the optimal value OBJ is at least 1 and the algorithm proceeds to set $k_{\max} = \min(\text{OBJ}, L)$, where L is the user's desired maximum number of covariates in the model. Then it proceeds to determine a set of Γ values to test. By default, the set is logarithmically spaced between 0 and $0.5\sqrt{p}$. This maximum value of Γ would likely force $\boldsymbol{\beta} = 0$ if the problem were completely unconstrained. At this point, the algorithm proceeds to Stage 2.

3.2 Stage 2: The MINLO Model

The heart of the method is following MINLO problem. We describe the MINLO here and devote Section 4.3 to explaining our techniques for solving it:

$$\begin{aligned} \text{(7a)} \quad \min_{\boldsymbol{\beta}, \mathbf{z}} \quad & \sum_{i=1}^{n} -y_i \left(\boldsymbol{\beta}' \mathbf{x}_i + (-1)^{y_i} \Gamma \|\boldsymbol{\beta}\|_{\frac{\alpha}{\alpha-1}}\right) \\ & + \log\left(1 + \exp\left(\boldsymbol{\beta}' \mathbf{x}_i + (-1)^{y_i} \Gamma \|\boldsymbol{\beta}\|_{\frac{\alpha}{\alpha-1}}\right)\right), \end{aligned}$$

$$\text{(7b)} \quad \text{s.t.} \quad z_\ell \in \{0,1\}, \quad \ell = 1, \ldots, p,$$

$$\text{(7c)} \quad -\mathcal{M} z_\ell \le \beta_\ell \le \mathcal{M} z_\ell, \quad \ell = 1, \ldots, p,$$

$$\text{(7d)} \quad \sum_{\ell=1}^{p} z_\ell \le k,$$

$$\text{(7e)} \quad z_1 = \cdots = z_\ell, \quad (1, \ldots, \ell) \in \mathcal{GS}_m \ \forall m,$$

$$\text{(7f)} \quad z_i + z_j \le 1 \quad \forall (i,j) \in \mathcal{HC},$$

$$\text{(7g)} \quad \sum_{i \in \mathcal{T}_m} z_i \le 1 \quad \forall m,$$

$$\text{(7h)} \quad z_\ell = 1 \quad \forall \ell \in \mathcal{I},$$

$$\text{(7i)} \quad \sum_{\ell \in \mathcal{S}_i} z_\ell \le |\mathcal{S}_i| - 1 \quad \forall \mathcal{S}_1, \ldots, \mathcal{S}_j.$$

In the objective function (7b), the robustification parameter Γ immunizes the resulting model against structural uncertainty in the data. In Constraint (7a), a binary indicator variable z_ℓ is introduced for every β_ℓ in the model. For a large enough constant $\mathcal{M}$, the constraint (7c) ensures that β_ℓ will only be included in the model if $z_\ell = 1$. The constraint (7d) limits the number of total variables that will be included in the model. This ensures general sparsity of the resulting model. The constraints in (7e), (7f) and (7g) are selective sparsity constraints. For the mth set of variables with a group sparsity structure, the set of constraints defined in (7e) ensures that the variables in $\mathcal{GS}_m$ are either all zero, or all nonzero. The set of constraints in (7f) ensure that the resulting model is free from extreme pairwise multicollinearity. The set $\mathcal{T}_m$ refers to the mth variable which was flagged as a candidate for transformation and all of its possible nonlinear transformations. The set of constraints (7g) ensures that at most one of the variables from the set $\mathcal{T}_m$ will be included in the final model for each of the candidate variables m. If $\mathcal{I} \neq \varnothing$, Constraint (7h) will be included in the model and will ensure that each of the specified independent variables appears in the final model. (7i) is a set of constraints to exclude particular solutions $\mathcal{S}_i$, such as those with high global multicollinearity or containing variables which are statistically insignificant. $\mathcal{S}_i$ is the set of indices corresponding to nonzero $\boldsymbol{\beta}$ value in the ith solution. The initial MINLO model will not contain line (7i); these constraints will be generated in Stage 3, if necessary.

The algorithm described in Section 4 is used to solve Problem (7) for each value of k from 1 to $k_{\max}$ and each value of Γ using the training data $\mathbf{y}$ and $\mathbf{X}$. For each problem solved, the output of the MINLO is a

set of variables $\boldsymbol{\beta}^*$ and $\boldsymbol{z}^*$. We measure and record the out-of-sample AUC on the validation set using this $\boldsymbol{\beta}^*$. Once the MINLO model is run for all potential values of k and Γ, the algorithm chooses the three sets of $\boldsymbol{\beta}$ with the highest AUC on the validation set as the top three regression models, and proceeds to Stage 3.

3.3 Stage 3: Generating Additional Constraints

We denote the top three sets of $\boldsymbol{\beta}$ by $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$. For each of the sets $\mathcal{S}_i$, the algorithm computes the significance levels for each of the variables via bootstrap methods. We note that our approach may be subject to the post-model selection inference problem of increased type 1 error, since the training data was used both to generate a candidate set of models and to generate an empirical distribution for each coefficient estimate. Post-model selection inference without type 1 error is an area of current research (see, e.g., [4, 25, 40]). Remedies suggested in the literature include reserving some of the original data purely for inference, or applying new theories of calculating and sizing confidence intervals. These methods can be incorporated within our approach.

Our algorithm also calculates the condition number of the model for each of the sets $\mathcal{S}_i$.

If a set $\mathcal{S}_i$ produces undesirable results—a condition number higher than desired, or a model with insignificant variables—the algorithm generates Constraint (7i) to exclude that set from the candidates of sets of best regression models.

Excluding set $\mathcal{S}_i$ can be achieved by "cutting off" the corner from the binary hypercube formed by the z variables using the constraint $\sum_{\ell \in \mathcal{S}_i} z_\ell \leq |S_i| - 1$. For example, to exclude set $\mathcal{S}_1 = \{3, 5, 11\}$, we can insert the constraint $z_3 + z_5 + z_{11} \leq 2$ into Problem (7) and resolve. The algorithm generates these additional constraints to exclude sets $\mathcal{S}_1, \ldots, \mathcal{S}_j$ as needed, and returns to Stage 2. The modeler may set the maximum condition number she will accept in the model, as well as the number of iterations she will permit between Stage 2 and Stage 3. The defaults are 30 and 3, respectively. In our experience, if a logistic regression model is a good fit for the data, few iterations are necessary.

When the algorithm ends, it presents the top three models, along with their condition numbers and confidence intervals of the bootstrapped coefficients.

Our MINLO approach can accommodate constraints beyond those specified in Problem (7). Our formulation assumes a robustified version of the traditional logistic regression goal, which is to minimize negative log likelihood. However, any of the other robust approaches to logistic regression mentioned in Section 2.3 could be substituted. Residual diagnostics could also be included, such as the calculation of Pearson residuals or deviance residuals. These can help the modeler test the validity of modeling using a logit function.

4. SOLVING THE MIXED INTEGER NONLINEAR OPTIMIZATION PROBLEM

In this section, we present a brief overview of mixed integer nonlinear optimization (MINLO) and explain our methodology for solving Problem (7).

4.1 Mixed Integer Optimization Landscape

The general form of a Mixed Integer Optimization (MIO) problem is as follows:

$$\begin{aligned} \min \quad & h(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & g_j(\boldsymbol{\alpha}) \leq 0 \quad \forall j \in J, \\ & \alpha_i \in \{0, 1\} \quad \forall i \in \mathcal{I}, \\ & \alpha_j \in \mathbb{R} \quad \forall j \notin \mathcal{I}, \end{aligned}$$

where $\mathbb{R}$ denotes the real numbers, the symbol $\leq$ denotes element-wise inequalities and we optimize over $\boldsymbol{\alpha} \in \mathbb{R}^m$ containing both discrete ($\alpha_i, i \in \mathcal{I}$) and continuous ($\alpha_i, i \notin \mathcal{I}$) variables, with $\mathcal{I} \subset \{1, \ldots, m\}$.

Types of MIO problems include mixed integer linear optimization (MILO) problems (h, g_j are linear functions), mixed integer quadratic optimization (MIQO) problems (h is quadratic, g_j are linear functions), and mixed integer nonlinear optimization (MINLO) problems (h and g_j are continuously differentiable nonlinear functions). When $\mathcal{I} = \varnothing$, MILO problems reduce to linear optimization (LO) problems, MIQO problems reduce to quadratic optimization (QO) problems, and MINLO problems reduce to nonlinear optimization (NLO) problems.

From 1991–2015, the overall speedup of MILO solvers was a factor of 780,000 and of hardware was a factor of 580,000, leading to an overall speedup of approximately 450 billion. Problem (7) is a MINLO, which is more challenging. However, we will make use of MILO solvers in our approach to solve the MINLO to optimality. MIO solvers provide both feasible solutions as well as lower bounds to the optimal value. As the MIO solver progresses toward the optimal solution, the lower bounds improve and provide an increasingly better guarantee of suboptimality, which is especially useful if the MIO solver is stopped before reaching the global optimum. In contrast, heuristic methods do not provide such a certificate of suboptimality.

The belief that MIO approaches to problems in statistics are not practically relevant was formed in the 1970s and 1980s and it was at the time justified. Given the astonishing speedup of MIO solvers and computer hardware in the last twenty-five years, the mindset of MIO as theoretically elegant but practically irrelevant is no longer justified. In this paper, we provide empirical evidence of this fact in the context of building a high-quality logistic regression model.

Developing algorithms for solving convex MINLO problems to provable optimality has been an active area of research since the 1970s, and a wide variety of MINLO solvers have been built based on these algorithms. Such algorithms integrate techniques from nonlinear optimization, integer optimization and linear optimization. Typically, these algorithms rely on two major solution techniques: (1) branch and bound with nonlinear relaxations or (2) linear relaxations of h and g_j. For background on convex MINLO, see [11]. See [13] for a complete categorization of twenty-four existing MINLO solvers.

4.2 Computational Tests on Existing MINLO Solvers

In this section, we examine whether current state of the art MINLO solvers are adequate for solving (2) and (7) in order to assess the need to develop a specialized solver.

We built six test problems. in order to compare MINLO solver performance. $\mathbf{x}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma}), i = 1, \ldots, n$ were independent realizations from a p-dimensional multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma} := (\sigma_{ij})$. The columns of the $\mathbf{X}$ matrix were standardized such that the training set had columns with zero mean and unit ℓ_2-norm. For a fixed $\mathbf{X}_{n\times p}$, we generated the response $\mathbf{y}$ as follows: $\mathbf{y}_i = \mathrm{Round}(1/(1 + \exp(-\boldsymbol{\beta}'\mathbf{x}_i + \varepsilon_i)))$, where $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. We denote the number of nonzeros in $\boldsymbol{\beta}$ by k. In particular, we took $\sigma_{ij} = \rho^{|i-j|}$ for $i, j \in \{1, \ldots, p\} \times \{1, \ldots, p\}$. In our experiments, we consider $k = 5$ and $\beta_i = 1$ for $i \in \{1, \ldots, p\}$ such that $i \bmod p/k = 0$ to generate k equally spaced values.

We restricted the first three of the test problems to Problem (2). For the second three test problems, we considered Problem (7) with general sparsity constraints, robustness in the objective function and pairwise multicollinearity constraints for any pair of covariates with correlation over 0.7. The exact parameters (n, p, ρ) of each of the six test problems were as follows. Problem 1: (100, 10, 0.4), Problem 2 (1000, 100, 0.4), Problem 3: (2000, 200, 0.4), Problem 4: (100, 10, 0.8), Problem 5: (1000, 100, 0.8), Problem 6: (2000, 200, 0.8). All problems were tested with $k = 5, \sigma = 2$.

The NEOS server makes many optimization solvers freely available for use on their servers [18, 20, 29]. By using the NEOS server, we were able to test six different solvers side by side using an AMPL interface The server "neos-6," where our computational experiments were performed, has the following computational specifications: 2.2 GHz processor, 24 cores, 64 GB of RAM and 2 TB of hard disk space. We tested all six solvers for which an AMPL interface was available: Bonmin, KNITRO, FilMINT, MINLP, SCIP and Couenne (see [13] for details on these solvers). We did not change the default options on any of the solvers.

Table 2 presents a comparison of times (in seconds) for each solver to reach optimality on each test problem, up to a maximum cut off time of 7200 seconds (2 hours). Note that we did not solve each test problem for every value of k, but only for the value of k corresponding to the true value of k. Thus, the times presented are the solve times for a single instance of the problem, averaged over five runs. We do not present

TABLE 2
MINLO solver comparison times (in seconds)

Solver	Pr. 1	Pr. 2	Pr. 3	Pr. 4	Pr. 5	Pr. 6
Bonmin	11	168	2370	14	Failed	Failed
KNITRO	16	29	145	0.42	6585	Failed
FilMINT	10	1283	633	Failed	Failed	Failed
MINLP	11	≈300*	≈6000*	10	≈6000*	Failed
SCIP	Cut off	Cut off	Cut off	Cut off	Cut off	Cut off
Couenne	Cut off	Cut off	Cut off	Cut off	Cut off	Cut off

*Note that the MINLP solver did not provide timestamps for solve times beyond 1 minute so these are rounded times based on computer clock time.

the time results with the goal of accurately benchmarking the best time possible, but rather to give a sense of generally expected solve times under the standard conditions of operating on a shared server.

The solve times presented in Table 2 represent the time to build and solve the problem. Thus, if we can embed the solver within a optimization language such that the problem does not have to rebuilt for successive values of k, we can expect that subsequent re-solves for additional values of k would be much faster. With this in mind, the solve time of the first four solvers on the NEOS server for Problem 1 may be efficient enough for practical purposes to solve the best subset problem, especially since $p = 10$ in Problem 1. However, the increased complexity of Problems 2 and 3 indicated the variability between solvers—and the inability to scale effectively to problems of a typical size. Problems 4, 5 and 6 are more challenging, mainly due to the addition of robustness in the objective function. Indeed, only three of the six available solvers were able to solve Problem 4, and only two of the six were able to solve Problem 5. None of the available solvers were able to solve Problem 6. These challenges of scale and complexity provided the motivation to create our own tailored algorithm to solve Problem (7) efficiently.

4.3 Tailored Algorithm

We built a tailored algorithm to efficiently solve the MINLO model (7). This consisted of two main ingredients: outer approximation methods and dynamic constraint generation. In the special case where the MINLO model only contains general sparsity constraints and the problem reduces to the best subset problem in logistic regression, we add a third ingredient: a discrete first-order heuristic.

4.4 Outer Approximation Methods

The outer approximation algorithm for convex MINLO was introduced in [21]. The algorithm alternates between solving a mixed integer linear optimization problem and a pure nonlinear optimization problem, where linearizations of the objective function around solutions to the NLO are added to the MIO. These linearizations are obtained by the convexity and differentiability of f: for any value of $\hat{\boldsymbol{\beta}} \in \mathbb{R}$, the following linear inequality is valid: $f(\boldsymbol{\beta}) \geq f(\hat{\boldsymbol{\beta}}) + \nabla f(\hat{\boldsymbol{\beta}})'(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})$.

In our case, the algorithm proceeds as follows. First, Problem (1) is solved and has optimal solution $\boldsymbol{\beta}^{\mathrm{NLO}}$. The following MILO, which we call the reduced master problem (RMP), is formed:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}} \ & \eta \\
\text{s.t.} \ & \eta \geq f(\boldsymbol{\beta}^{\mathrm{NLO}}) + \nabla f(\boldsymbol{\beta}^{\mathrm{NLO}})'(\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{NLO}}) \\
& z_\ell \in \{0,1\}, \quad \ell = 1, \ldots, p, \\
& -\mathcal{M} z_\ell \leq \beta_\ell \leq \mathcal{M} z_\ell, \quad \ell = 1, \ldots, p, \\
& \sum_{\ell=1}^{p} z_\ell \leq k, \\
& z_1 = \cdots = z_\ell \quad (1, \ldots, \ell) \in \mathcal{GS}_m \ \forall m, \\
& z_i + z_j \leq 1 \quad \forall (i,j) \in \mathcal{HC}, \\
& \sum_{i \in \mathcal{T}_m} z_i \leq 1 \quad \forall m, \\
& z_\ell = 1 \quad \forall \ell \in \mathcal{I}, \\
& \sum_{\ell \in \mathcal{S}_i} z_\ell \leq |\mathcal{S}_i| - 1 \quad \forall \mathcal{S}_1, \ldots, \mathcal{S}_j.
\end{aligned}
\tag{8}
$$

The reduced master problem (8) is solved. The support of the resulting solution, $\boldsymbol{\beta}^{\mathrm{RMP}}$, is then fixed, and the following nonlinear optimization problem is solved:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}} \ & f(\boldsymbol{\beta}) \\
\text{s.t.} \ & \operatorname{support}(\boldsymbol{\beta}) = \operatorname{support}(\boldsymbol{\beta}^{\mathrm{RMP}}).
\end{aligned}
\tag{9}
$$

The solution to Problem (9) is a new $\boldsymbol{\beta}^{\mathrm{NLO}}$. Linearizations around this new $\boldsymbol{\beta}^{\mathrm{NLO}}$ are added to the reduced master problem (8), and the algorithm continues to alternate between solving Problems (8) and (9). At each stage, these cutting plane linearizations cut off the current integer solution to Problem (8) unless the integer solution is optimal for Problem (7). As the algorithm progresses, the reduced master problem (8) becomes an increasingly closer approximation to Problem (7). The global minimum of Problem (7) is reached when the objective function of the reduced master problem (8) is within some pre-specified tolerance ε of the objective function of the NLO problem (9).

4.5 Dynamic Constraint Generation

We implement an efficient way to solve the reduced master problem (8). In general, outer approximation methods are known as "multi-tree" methods because every time a linearization is added, the reduced master problem (8) must be solved again. Over the course of

the solution process, multiple branch and bound trees are built in order to solve successive versions of the reduced master problem (8). We implement a "single-tree" way of solving Problem (8) by using dynamic constraint generation, known in the optimization literature as *lazy constraint callbacks*, which dynamically (or lazily) add cutting planes to the model whenever an integer feasible solution is found. Unless the current integer solution is optimal, this will refine the feasible region of the problem by cutting off the current integer solution.

Lazy constraint callbacks are a relatively new type of callback. CPLEX 12.3 introduced lazy constraint callbacks in 2010 and Gurobi 5.0 introduced lazy constraints in 2012. To date, the only MIO solvers which provide lazy constraint callback functionality are CPLEX [33], Gurobi [34] and GLPK [26]. The outer approximation method for solving convex MINLO does not require lazy constraint callbacks, but if we do exploit their functionality, only one branch and bound tree needs to be built. This saves the rework of rebuilding a new branch and bound tree every time a new integer feasible solution is found in Problem (8).

Lazy constraints are a fairly new feature within optimization solvers. Although many problems within statistics are naturally formulated as MIO or MINLO problems, to the best of our knowledge, we are the first to integrate the optimization-based concept of lazy constraints into the process of building a statistical model.

4.6 A Discrete First-Order Heuristic for Best Subset Selection

In the case where we are interested in Problem (2), the classical best subset problem in logistic regression, we add a third ingredient to our tailored algorithm: a heuristic for solving (2) based on a discrete extension of first-order methods in convex optimization.

We first developed this heuristic in [8] and repeat the approach in the supplemental files for clarity. This gives near-optimal solutions to Problem (2), and we incorporate this solution at the beginning of our tailored algorithm by adding a linearization around the solution to Problem (9) with $\boldsymbol{\beta}^{\mathrm{RMP}}$ taken as the first-order heuristic solution. By doing this in the very first step, we ensure that a high-quality cutting plane is added immediately to Problem (8), causing the outer approximation algorithm to converge much more quickly.

5. COMPUTATIONAL RESULTS

The computational tests in Section 5.1 were performed on a computer with an Intel Xeon E5440 (2.8 GHz) processor with 8 cores and 32 GB of RAM in order to fairly compare times with the NEOS server. All other computational tests were performed on a computer with an Intel Xeon E5687W (3.1 GHz) processor, 16 cores and 128 GB of RAM. We used `Gurobi 6.0.0` [34] as the optimization solver, and implemented the algorithm in `Julia 0.3.3` [9], a technical computing language. We used `JuMP 0.7.0` [41], an algebraic modeling language package for Julia, to interface with Gurobi. We used the `GLMNet 0.0.2` package in Julia to compute Lasso solutions.

5.1 Time Comparision

We begin by comparing our algorithm's performance to the same set of six test problems from Section 4.2, again averaging times over five trials (see Table 3).

In these trials, our tailored algorithm was uniformly faster than the six optimization solvers we tested on the NEOS server. Moreover, this speed comparison indicates that our algorithm can scale to higher dimensional problems more easily than other existing MINLO software, and can handle the increased challenges of solving Problems 4–6. Next, we compare the performance of solving Problem (7) using MINLO compared to heuristic methods. Again, we consider problems restricted to general sparsity constraints, and problems requiring other statistical properties as well.

5.2 Methodology Comparison—Best Subset

As indicated in Section 2.1, logistic regression with an ℓ_1-penalty is the primary method for inducing sparsity in logistic regression models. In this section, we compare our methodology for the best subset problem with Lasso for logistic regression and report sparsity and predictive performance. We measure predictive performance using area under the ROC curve (AUC) as our metric. Data was generated as per Section 4.2.

TABLE 3
MINLO solver comparison times (in seconds)

Solver	Pr. 1	Pr. 2	Pr. 3	Pr. 4	Pr. 5	Pr. 6
OURS	<1	15	16	<1	155	258

*Note that the MINLP solver did not provide timestamps for solve times beyond 1 minute so these are rounded times based on computer clock time.

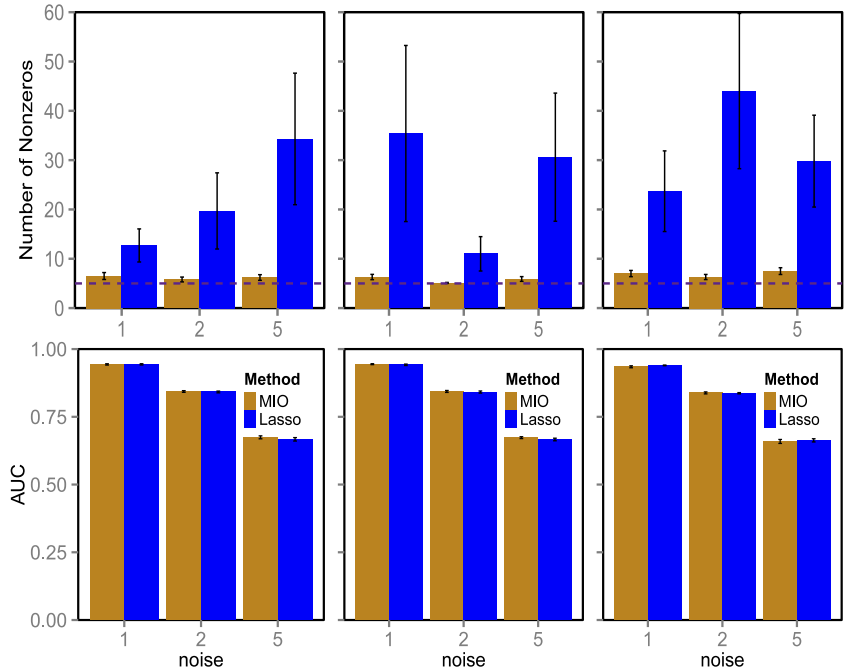

FIG. 1. *Series of computational tests for Problem* 2 *with* $n = 2000$, $p = 200$. *The left panel is* $\rho = 0$, *the middle panel is* $\rho = 0.4$, *and the right panel is* $\rho = 0.8$. *The dashed line in the top panel represents the true number of nonzero values. Error bars represent standard errors.*

Overdetermined regime. We begin by considering the traditional overdetermined regime with $n > p$. Figure 1 shows a representative case within the overdetermined regime with $n = 2000$ and $p = 200$. We note that the MINLO approach and the Lasso approach perform almost identically with respect to AUC across many different noise (σ) and correlation (ρ) levels. Where we notice a large difference between the two methods is in the number of nonzero coefficients chosen by the two methods. MINLO significantly outperforms Lasso in this respect. In this example, there are five true nonzero coefficients. MINLO never selects more than seven. Lasso selects far more variables to enter the model, and is less consistent than MINLO: we observe far greater standard error over the ten trials.

High dimensional regime. Our method is applicable both in the traditional overdetermined $n > p$ regime and in the increasingly common high dimensional underdetermined $n < p$ regime. The tailored approach of using mixed integer optimization in conjunction with warm starts and lazy constraint cutting planes generated by pure nonlinear optimization rapidly finds the optimal solution.

However, in the $n < p$ regime, we observe that the lower bounds of the mixed integer optimization problem progress slowly, so while the optimal solution may have been found, certification of optimality happens slowly, if at all.

To address this, we follow the approach of [8] and consider adding bounding box constraints to the MINLO formulation. These constraints limit the search space, and allow the solver to certify optimality within the bounding box. In particular, we consider the following additional bounding box constraints to the reduced master problem (8):

$$\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \leq \mathcal{L}^{\beta}_{\ell,\mathrm{loc}},$$

where $\boldsymbol{\beta}_0$ is a candidate sparse solution. The radius of the ℓ_1-ball above, that is, $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}}$, is a user-defined parameter which controls the size of the feasible set.

In our experiments, we ran our tailored algorithm for 180 seconds, and used the resulting solution as $\boldsymbol{\beta}_0$. We then generated the box constraint using $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}} = \|\boldsymbol{\beta}_0\|_1 / k$. Figure 2 gives sparsity and predictive performance results for an example in the high dimensional regime with $n = 400$ and $p = 1000$.

We notice that in this example, Lasso frequently, but not always, has slightly better predictive performance than MINLO. Nevertheless, the number of nonzero coefficients chosen by Lasso are far higher than the number selected by MINLO. MINLO consistently chooses a number of nonzeros in the neighborhood of the true

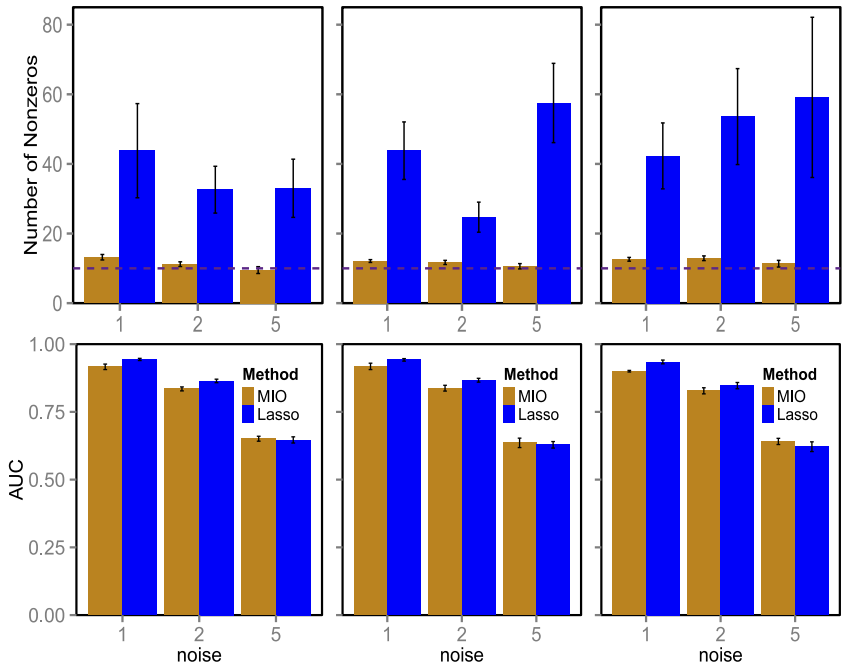

FIG. 2. *Series of computational tests for Problem* 4 *with* $n = 400$, $p = 1000$. *The left panel is* $\rho = 0$, *the middle panel is* $\rho = 0.4$ *and the right panel is* $\rho = 0.8$. *The dashed line in the top panel represents the true number of nonzero values. Error bars represent standard errors.*

number; the Lasso solution exhibits much higher standard error of the mean, and is usually 2–4 times the true number of nonzeros.

These observations about predictive performance and sparsity in the high dimensional regime are commensurate with the remarks in [8] that Lasso is a *robust* method first and foremost, and a sparsity-inducing method second. Even so, we doubt that the minor increase in predictive performance that Lasso's robustness may induce is worth the tradeoff of introducing so many variables into the model.

Extremely high dimensionality. The recent data explosion has led statisticians to face problems of extremely high dimensionality. One place where this arises frequently is in the area of gene expression data. To show that our approach is relevant in the extremely high dimensional regime, we consider an example using data generated by the The Cancer Genome Atlas Research Network: http://cancergenome.nih.gov/. Tumor formation is a result of dysfunctional proteins and/or signaling network in cells. Often, tumor types have distinct subtypes which each have their own genetic signatures. Here, we will consider two different subtypes of lung cancer: adenocarcinoma and squamous cell carcinoma, and use best subset logistic regression to identify which genes signal each of the two subtypes.

The dataset we consider has gene expression data for 992 patients, as well as a binary variable indicating that patient's cancer subtype (50.7% adenocarcinoma, 49.2% squamous cell). The original gene expression data contained results for 30,373 genes. After removing all genes which had missing data for some patients, we had a dataset with $n = 992$ and $p = 13{,}937$.

We divided the dataset into training, validation, and test sets with the ratio of 50%, 25%, 25%. We ran Lasso for logistic regression on the training set and, using the model with the best validation set AUC, determined a subset of important genes. We repeated this approach ten times with different random splits of the data in order to generate a full set of important genes: this yielded 300 genes. The average AUC of the Lasso model over the ten trials was 0.977 ± 0.02.

We used our tailored method on this reduced dataset of 300 genes. Over ten trials, the MINLO approach selected a total of 120 genes, with an average AUC of 0.973 ± 0.01. The two subtypes are clinically dissimilar, so even with such a high AUC, these models are unlikely to be helpful for diagnosis. However, the lower number of genes selected by MINLO may be very helpful in identifying biomarkers that are essential for monitoring drug effects in clinical trials, and for helping researchers understand cell signaling perturbation in different tumor subtypes so as to direct drug discovery and development efforts.

TABLE 4
Pairwise multicollinearity; $n = 1000$, $p = 100$, *true* $K = 5$, $\rho = 0.9$, $\mathbf{\Delta X} = 0$

		MINLO							Lasso				
ρ	σ	Γ^*	K^*	TP	AUC	MC	Con	Time	K^*	TP	AUC	MC	Con
0.9	1	0.000	4.6	4.6	0.941	0.163	4.8	5606	23.9	5.0	0.938	0.904	84.6
		0.000	0.2	0.2	0.003	0.007	1.3	470	5.7	0.0	0.002	0.002	21.5
0.9	2	0.000	4.8	4.4	0.839	0.223	2.4	4859	20.5	4.9	0.834	0.870	69.7
		0.000	0.2	0.2	0.004	0.053	0.6	583	5.3	0.1	0.006	0.019	21.7
0.9	5	0.001	4.8	2.6	0.658	0.296	1.7	4778	27.2	3.8	0.658	0.824	109.1
		0.000	0.3	0.3	0.004	0.064	0.4	462	8.8	0.3	0.005	0.077	44.7

5.3 Methodology Comparison—Full Algorithmic Approach

Our main goals of the algorithmic approach to logistic regression are to achieve interpretability and robustness, while retaining predictive power.

First, we present results in Tables 4 and 5 for synthetic datasets for the default parameters of the algorithm: five values of Γ tested and 0.7 as the maximum pairwise correlation allowed. These are designed to illustrate the algorithmic's approach ability to handle datasets with high multicollinearity and to be robust against added noise. Then we tested our algorithm on five publicly-available real datasets and present these results in Table 6. Finally, we consider a combined synthetic example designed to demonstrate the capacity of the algorithmic approach to identify various properties when presented in concert. Note that in all cases, all variables selected by the algorithmic approach are significant at the 0.05 level.

Preliminaries. Each experiment corresponds to two rows in a table. The top row presents average results over ten trials of the same experiment and the bottom row presents the standard error. We use the following notation: $K^* =$ value of k chosen by the algorithm, TP = number of true nonzero variables identified by the algorithm, for the synthetic datasets, MC = the maximum pairwise correlation present in the final model and Con = condition number. Time for the MINLO algorithm is presented in seconds, and is not meant to accurately benchmark the best possible time but to show that it is computationally tractable to solve these problems in a practical amount of time on standard computers. The real datasets were obtained from [2]. We abbreviate each real dataset's name as follows: "Bank" stands for the Banknote Authentication dataset; "Telescope" corresponds to the Magic Gamma Telescope dataset; "Mass" stands for the Mammographic Mass dataset; "Ozone 8" corresponds to the Ozone Detection Level Eight dataset; and "Ozone 1" stands for the Ozone Detection Level One dataset. We aim to return solutions in practical amounts of time, so we imposed a 60-second time limit on each optimization problem solved. Often optimality is reached before the time limit. Note that for each dataset, $K_{\max} \times$ (# of values of Γ tested) $\times$ (# of iterations of Stage 3) MINLO problems are solved.

Results. Table 4 shows results for synthetic logistic regression datasets with high pairwise multicollinearity. We observe that the MINLO model achieves the

TABLE 5
Robustness; $n = 1000$, $p = 100$, *true* $K = 5$, $\rho = 0$, $\mathbf{\Delta X} \sim \text{Uniform}(0, 2)$

		MINLO							Lasso				
ρ	σ	Γ^*	K^*	TP	AUC	MC	Con	Time	K^*	TP	AUC	MC	Con
0	1	0.0000	5.1	5.0	0.873	0.057	1.2	1196	27.5	5.0	0.867	0.088	1.8
		0.0000	0.1	0.0	0.004	0.005	0.0	26	7.3	0.0	0.005	0.007	0.2
0	2	0.0002	5.2	5.0	0.793	0.059	1.2	989	19.9	5.0	0.788	0.088	1.6
		0.0001	0.2	0.0	0.007	0.004	0.0	29	5.4	0.0	0.008	0.009	0.1
0	5	0.0000	5.3	4.8	0.655	0.064	1.2	1027	17.4	5.0	0.641	0.091	1.6
		0.0000	0.2	0.1	0.007	0.005	0.0	14	4.6	0.0	0.008	0.005	0.1

TABLE 6
Results for real datasets

			MINLO					Lasso			
Dataset	***n***	***p***	K^*	**AUC**	**MC**	**Con**	**Time**	K^*	**AUC**	**MC**	**Con**
Bank	686	4	2.9	0.956	0.360	3.8	4.1	3.9	0.994	0.783	12.1
			0.1	0.002	0.011	0.2	1.0	0.1	0.000	0.003	0.4
Telescope	9510	10	4.8	0.832	0.668	7.2	1145.3	3.2	0.822	0.272	2.7
			0.2	0.002	0.027	0.8	166.0	0.2	0.002	0.068	0.8
Mass	415	10	4.8	0.875	0.406	6.3	24.0	6.2	0.873	0.434	9.5
			0.6	0.007	0.016	1.3	3.8	0.8	0.006	0.019	2.3
Ozone 8	924	72	3.1	0.869	0.283	2.6	1583.1	38.1	0.895	0.982	9293.8
			0.3	0.005	0.047	0.4	271.2	3.6	0.007	0.010	2827.0
Ozone 1	924	72	6.5	0.885	0.644	20.0	725.5	38.9	0.888	0.984	12,283.9
			0.8	0.013	0.026	4.9	122.9	4.8	0.016	0.010	4869.6

same, or slightly higher, AUC than Lasso. The MINLO model performs better in terms of sparsity, however, as noise increases, this is at the expense of recovering the true set of nonzero coefficients. However, the final Lasso models contain very high pairwise collinearity and condition numbers that indicate severe multicollinearity issues.

Table 5 shows results for datasets designed to illustrate robustness. The MINLO model achieves very slightly better predictive power than the Lasso model. In the highest noise setting ($\sigma = 5$), MINLO does not always fully recover the true set of nonzero coefficients. Nevertheless, the proportion of coefficients selected that are truly nonzero remains quite high on average ($4.8/5.3 = 90.6\%$) compared to Lasso ($5.0/17.4 = 28.7\%$).

We tested our algorithm on five publicly-available real datasets and present these results in Table 6. Note that n here indicates the size of the training dataset—the original dataset has $2n$ observations.

In the Banknote Authentication dataset, MINLO achieves slightly better sparsity, but slightly worse AUC; this is the price of interpretability, since with a threshold of 0.7 as the maximum pairwise correlation, the MINLO model cannot include as many variables as the Lasso model. In the Magic Gamma Telescope dataset, however, we see the opposite result: Lasso outperforms MINLO with respect to sparsity, at the expense of AUC. MINLO achieves a slightly higher AUC within the bounds of a 0.7 maximum pairwise correlation limit. It is not surprising that the algorithmic approach trades off the desirable properties of low multicollinearity, sparsity and predictive performance in different ways for different datasets. In fact, what we can be assured of is that the MINLO model trades these properties off in an optimal way given the constraints the modeler specifies. It is likely that if a lower maximum correlation threshold were given, the Magic Gamma Telescope results would show a lower K^* selected—but possibly a lower test set AUC as well. We verified this intuition by running the Magic Gamma Telescope dataset again with a maximum pairwise correlation threshold of 0.5—results are in Table 7. Likewise, were a higher maximum correlation threshold specified, it is likely that the Banknote Authentication test set AUC would match Lasso's—but with a higher pairwise correlation, and higher condition number. The Mammographic Mass dataset is an example where the MINLO approach outperforms Lasso on all levels: a lower K^* selected, higher test set AUC, and lower maximum correlation and condition number. The Ozone Detection datasets both have a much greater number of potential variables than the other three datasets. As we have come to expect in such cases, the MINLO model significantly outperforms Lasso with respect to sparsity here. Predictive performance is similar, although slightly lower in the MINLO case. However, the resulting maximum collinearity is drastically improved in the MINLO model.

Finally, we consider a combined synthetic example which we created in order to test the algorithm's ability to identify many properties when presented together. Specifically, we consider an example whose structure incorporates general sparsity, selective sparsity in terms of both high pairwise multicollinearity and group sparsity, and modeler expertise in a single dataset. We test this example in the case where $n = 2000$ and $p = 200$.

TABLE 7
Magic Gamma Telescope results with maximum pairwise correlation threshold of 0.5

MINLO					Lasso			
K^*	AUC	MC	Con	Time	K^*	AUC	MC	Con
3.3	0.815	0.232	1.9	128.8	3.0	0.819	0.184	1.6
0.2	0.002	0.029	0.2	17.0	0.0	0.003	0.002	0.0

We generated a synthetic data matrix $\mathbf{X}$ for $n = 2000$, $p = 100$ according to the process outlined previously. We used a value of $\rho = 0.9$ to ensure that there is high pairwise multicollinearity present between some columns of $\mathbf{X}$, and $\sigma = 5$ to ensure high noise. To generate nonlinear transformations, for each column j of $\mathbf{X}$ we included an additional column consisting of the squared entries of j, bringing the total number of potential covariates up to 1000. We consider $k = 10$. However, we generated $\beta_i = 1$ so that 7 positive values occurred in the original 100 columns and 3 were located in the 100 transformed columns. The response $\mathbf{y}$ was generated as before as $\mathbf{y}_i = \text{Round}(1/(1 + \exp(-\boldsymbol{\beta}'\mathbf{x}_i + \varepsilon_i)))$, where $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. To test our robustness to error in data, we generated a matrix $\Delta\mathbf{X} \sim \text{Unif}(0, 2)$ and considered $\mathbf{X} + \Delta\mathbf{X}$. We assume the modeler has some expertise with this sort of data, and knows one of the values of i such that β_i is truly nonzero. Finally, the modeler is also aware of a group sparsity structure and knows that $\beta_a, \beta_b, \beta_c$ and β_d are all either all zero or all nonzero and that $\beta_e, \beta_f, \beta_g$, and β_h are either all zero or all nonzero, where $\{a, b, c, d\} \in \{i \mid \beta_i = 1\}$ and $\{e, f, g, h\} \in \{i \mid \beta_i = 0\}$.

Table 8 presents results for this combined example. As before, the top row presents average results over five trials of the same experiment and the bottom row presents the standard error.

In this combined example, we see that MINLO produces a much lower total number of variables and lower pairwise multicollinearity and condition number while maintaining a similar test set AUC to the Lasso model. Although the true positive rate is lower for MINLO than Lasso in this challenging case, the precision (ratio of number of true positives chosen to total number of variables chosen) is much higher for MINLO.

The general pattern that these computational experiments of our algorithmic approach to logistic regression is that MINLO and Lasso typically exhibit very similar predictive performance. However, MINLO is frequently able to reduce the number of variables selected and/or reduce the multicollinearity in the model. The balance between these properties depends on the modeler's own input to the MINLO model.

5.4 Generality of Proposed Approach

We have thus far limited ourselves to synthetic datasets generated as per Section 4.2, and compared our results to Lasso. In practice, data could be generated in any number of ways, and results could be tested against any number of alternative algorithms.

We demonstrate the generality of our proposed approach in this section in two ways. First, we consider synthetic data generated as in Section 4.2, with the important distinction that we consider generating the data with nonzero coefficients which do not have equal absolute value. The equal absolute value setting is rare in practice, so showing that our methodology performs well in the unequal setting is critical.

Second, we consider alternative algorithms to Lasso. Lasso has often been criticized for selecting only moderately sparse solutions. Other heuristics may give

TABLE 8
Results for combined example

MINLO							Lasso				
Γ^*	K^*	TP	AUC	MC	Con	Time	K^*	TP	AUC	MC	Con
0.0004	11.2	6.2	0.76	0.56	5.9	1761.3	64.2	9.2	0.77	0.71	32.7
0.0002	0.6	0.6	0.00	0.03	0.5	40.9	6.9	0.3	0.00	0.00	1.1

TABLE 9
Unequal predictor coefficients—best subset problem

					MINLO			Lasso	
n	p	ρ	σ	SNR	K^*	AUC	Time	K^*	AUC
2000	200	0.4	2	1.12	6.10	0.909	539	19.2	0.910
					0.67	0.003	11.0	8.47	0.002
400	1000	0.4	5	0.45	8.30	0.689	4301	28	0.703
					0.83	0.019	60.6	5.02	0.013

sparser solutions while retaining comparable predictive power. Indeed, there are a suite of two-stage methods which use Lasso as one component of a sparse selection algorithm: for example, forward stepwise or forward stagewise logistic regression on the Lasso variables, adaptive lasso for logistic regression, and the Lasso MLE hybrid.

We expect most practitioners will turn to simpler heuristics like Lasso, since code for performing Lasso logistic regression is readily available and the method is well known. Nevertheless, we consider it important to test our method against other state-of-the-art algorithms, so in this section we test adaptive Lasso and display the results.

These tests aim to show that our method is widely applicable to a variety of input data and compares favorably to modern algorithms for inducing sparsity in logistic regression models.

Unequal predictor coefficients. We generated data according to Section 4.2 with $k = 5$ and $\beta_i \in \{-1, 2, 0.5, -2, 1.5\}$ for $i \in \{1, \ldots, p\}$ such that $i \bmod p/k = 0$ to generate k equally spaced values.

Table 9 shows sample results for the pure subset selection problem using this set of true β values for both the overdetermined regime and the high-dimensional regime. Table 10 shows sample results for our full algorithmic approach using this set of true β values. The experiments in Table 10 corresponds to the robustness and pairwise multicollinearity studies, respectively, from Section 5.3. Table notation and formatting is as in Section 5.3.

Adaptive Lasso. Adaptive Lasso was introduced in [55]. In this modification of Lasso, adaptive weights are used for penalizing the coefficients. When considering testing algorithms which are enhancements to Lasso, we choose to focus on adaptive Lasso for logistic regression since adaptive Lasso can be solved with a straightforward modification of the algorithm for solving the Lasso. Other techniques, be they modifications of Lasso or entirely different, do not have readily available code available for logistic regression.

We include a brief comparison of adaptive Lasso and Lasso performance on datasets generated according to Section 4.2. We also test the unequal coefficient case of the previous subsection. As before, $k = 5$ in these experiments (see Table 11).

In general, adaptive Lasso produces models with equivalent predictive power to Lasso which are sparser. We note that the degree to which the models are sparser is higher in the case where we test basic over- and under-determined data. When we introduce uniform random noise (as for our robustness tests) or high pairwise multicollinearity, the adaptive Lasso does not have quite as much of an edge on Lasso.

TABLE 10
Unequal predictor coefficients—full algorithmic approach

			MINLO							Lasso				
ρ	σ	ΔX	Γ^*	K^*	TP	AUC	MC	Con	Time	K^*	TP	AUC	MC	Con
0.4	2	2	0	4.9	2	0.850	0.100	1.303	1561	30.4	2	0.850	0.314	3.412
			0	0.3	0	0.005	0.030	0.091	135	7.18	0	0.006	0.031	0.415
0.9	2	0	0	4.2	1.8	0.885	0.205	2.066	3524	20.3	2	0.887	0.853	73.4
			0	0.3	0.1	0.005	0.057	0.635	563	5.70	0	0.005	0.029	27.4

TABLE 11
Adaptive Lasso

							Adaptive Lasso		Lasso	
Case	n	p	ρ	ΔX	σ	**SNR**	K^*	**AUC**	K^*	**AUC**
Overdetermined regime with	2000	200	0.4	0	2	1.12	26.90	0.841	22	0.842
Equal predictor coefficients	0	0	0	0	0	0	7.49	0.004	6.8	0.00
Underdetermined regime with	400	1000	0.4	0	5	0.45	56.70	0.582	88	0.583
Equal predictor coefficients	0	0	0	0	0	0	19.95	0.012	28.1	0.02
Overdetermined regime with	2000	200	0.4	0	2	1.12	13.60	0.911	33	0.910
Unequal predictor coefficients	0	0	0	0	0	0	4.75	0.002	8.9	0.00
Underdetermined regime	400	1000	0.4	0	5	0.45	14.10	0.710	42	0.696
Unequal predictor coefficients	0	0	0	0	0	0	6.33	0.013	24.6	0.02
Robustness with	1000	100	0.4	2	2	1.12	13.10	0.793	23	0.792
Equal predictor coefficients	0	0	0	0	0	0	3.42	0.006	5.4	0.01
Multicollinearity with	1000	100	0.9	0	2	1.23	22.00	0.845	21	0.844
Equal predictor coefficients	0	0	0	0	0	0	7.04	0.005	7.1	0.01
Robustness with	1000	100	0.4	2	2	1.12	19.50	0.836	22	0.834
Unequal predictor coefficients	0	0	0	0	0	0	4.85	0.003	6.6	0.00
Multicollinearity with	1000	100	0.9	0	2	1.23	11.90	0.883	17	0.884
Unequal predictor coefficients	0	0	0	0	0	0	2.48	0.005	3.7	0.00

Nevertheless, adaptive Lasso does not outperform MINLO methods, either in the pure subset selection case or the full algorithmic approach. See Sections 5.2 and 5.3 for corresponding MINLO results.

6. CONCLUSION

In this paper, we have developed a framework for creating logistic regression models with a wide variety of statistical properties. The core of this methodology is a MINLO model, and we develop a tailored algorithm to solve this challenging MINLO. This is the first algorithm that we are aware of to make use of callbacks within optimization software to solve a statistical problem. We have demonstrated that our algorithm converges to the optimal solution in faster times than existing off-the-shelf MINLO software.

Our approach is competitive with existing sparsity-inducing heuristics for logistic regression, namely, Lasso, with respect to predictive performance. Moreover, it frequently outperforms Lasso with respect to sparsity detection, and can guarantee many other desirable qualities in the model. We have demonstrated the effectiveness of this approach on real and synthetic datasets in producing high-quality logistic regression models within reasonable time frames.

SUPPLEMENTARY MATERIAL

Supplement to "Logistic Regression: From Art to Science" (DOI: 10.1214/16-STS602SUPP; .pdf).

REFERENCES

[1] BACH, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.* **9** 1179–1225. MR2417268

[2] BACHE, K. and LICHMAN, M. (2014). UCI machine learning repository. Available at http://archive.ics.uci.edu/ml. Accessed: 2014-08-20.

[3] BEN-TAL, A., EL GHAOUI, L. and NEMIROVSKI, A. (2009). *Robust Optimization*. Princeton Univ. Press, Princeton, NJ. MR2546839

[4] BERK, R., BROWN, L., BUJA, A., ZHANG, K. and ZHAO, L. (2013). Valid post-selection inference. *Ann. Statist.* **41** 802–837. MR3099122

[5] BERTSIMAS, D., BROWN, D. B. and CARAMANIS, C. (2011). Theory and applications of robust optimization. *SIAM Rev.* **53** 464–501. MR2834084

[6] BERTSIMAS, D., DUNN, J., PAWLOWSKI, C. and ZHUO, Y. D. (2017). Robust classification. *J. Mach. Learn. Res.* To appear.

[7] BERTSIMAS, D. and KING, A. (2017). Supplement to "Logistic Regression: From Art to Science." DOI:10.1214/16-STS602SUPP.

[8] BERTSIMAS, D., KING, A. and MAZUMDER, R. (2016). Best subset selection via a modern optimization lens. *Ann. Statist.* **44** 813–852. MR3476618

[9] BEZANSON, J., KARPINSKI, S., SHAH, V. B. and EDELMAN, A. (2012). Julia: A fast dynamic language for technical computing. Preprint. Available at https://arxiv.org/abs/1209.5145.

[10] BIANCO, A. M. and YOHAI, V. J. (1996). Robust estimation in the logistic regression model. In *Robust Statistics, Data Analysis, and Computer Intensive Methods* (*Schloss Thurnau*, 1994). *Lect. Notes Stat.* **109** 17–34. Springer, New York. MR1491394

[11] BONAMI, P., KILINÇ, M. and LINDEROTH, J. (2012). Algorithms and software for convex mixed integer nonlinear programs. In *Mixed Integer Nonlinear Programming* 1–39. Springer, Berlin.

[12] BOX, G. E. P. and TIDWELL, P. W. (1962). Transformation of the independent variables. *Technometrics* **4** 531–550. MR0184313

[13] BUSSIECK, M. R. and VIGERSKE, S. (2010). *Minlp Solver Software*. In *Wiley Encyclopedia of Operations Research and Management Science*. Wiley Online Library.

[14] CARROLL, R. J. and PEDERSON, S. (1993). On robustness in the logistic regression model. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **55** 693–706. MR1223937

[15] CHATTERJEE, S., HADI, A. S. and PRICE, B. (2012). *Regression Analysis by Example*, 5th ed. Wiley, New York.

[16] CRAMER, J. S. (2002). The origins of logistic regression. Technical report, Tinbergen Institute.

[17] CROUX, C. and HAESBROECK, G. (2003). Implementing the Bianco and Yohai estimator for logistic regression. *Comput. Statist. Data Anal.* **44** 273–295. Special issue in honour of Stan Azen: a birthday celebration. MR2020151

[18] CZYZYK, J., MESNIER, M. P. and MORÉ, J. J. (1998). The neos server. *J. Comput. Sci. Eng.* **5** 68–75.

[19] DOBSON, A. J. and BARNETT, A. G. (2008). *An Introduction to Generalized Linear Models*, 3rd ed. CRC Press, Boca Raton, FL. MR2459739

[20] DOLAN, E. D. (2001). Neos server 4.0 administrative guide. Preprint. Available at arXiv:cs/0107034.

[21] DURAN, M. A. and GROSSMANN, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36** 307–339.

[22] EFRON, B. (1979). Bootstrap methods: Another look at the jackknife. *Ann. Statist.* **7** 1–26. MR0515681

[23] ELDAR, Y. C. and KUTYNIOK, G. (2012). *Compressed Sensing*: *Theory and Applications*. Cambridge Univ. Press, London.

[24] FIGUEIREDO, M. A. T. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **25** 1150–1159.

[25] FITHIAN, W., SUN, D. and TAYLOR, J. (2014). Optimal inference after model selection. Preprint. Available at arXiv:1410.2597.

[26] FREE SOFTWARE FOUNDATION (2015). GNU linear programming kit. Available at http://www.gnu.org/software/glpk/glpk.html. Accessed: 2015-03-06.

[27] FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33** 1–22.

[28] FURNIVAL, G. M. and WILSON, R. W. (1974). Regressions by leaps and bounds. *Technometrics* **16** 499–511.

[29] GROPP, W. and MORÉ, J. (1997). Optimization environments and the neos server. In *Approximation Theory and Optimization* 167–182. Cambridge Univ. Press, Cambridge, UK.

[30] HILBE, J. M. (2011). *Logistic Regression Models*. CRC Press, Boca Raton, FL.

[31] HOSMER, D. W., JOVANOVIC, B. and LEMESHOW, S. (1989). Best subsets logistic regression. *Biometrics* **45** 1265–1270.

[32] HOSMER JR., D. W. and LEMESHOW, S. (2013). *Applied Logistic Regression*. Wiley, Hoboken, NJ.

[33] IBM ILOG CPLEX OPTIMIZATION STUDIO (2015). Cplex optimizer. Available at http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html. Accessed: 2015-03-06.

[34] GUROBI INC. (2014). Gurobi optimizer reference manual. Available at http://www.gurobi.com. Accessed: 2014-08-20.

[35] KIM, Y., KIM, J. and KIM, Y. (2006). Blockwise sparse regression. *Statist. Sinica* **16** 375–390. MR2267240

[36] KOH, K., KIM, S.-J. and BOYD, S. P. (2007). An interior-point method for large-scale l_1-regularized logistic regression. *J. Mach. Learn. Res.* **8** 1519–1555. MR2332440

[37] KRISHNAPURAM, B., CARIN, L., FIGUEIREDO, M. A. T. and HARTEMINK, A. J. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Mach. Intell.* **27** 957–968.

[38] KRISHNAPURAM, B., HARTERNINK, A. J., CARIN, L. and FIGUEIREDO, M. A. T. (2004). A Bayesian approach to joint feature selection and classifier design. *IEEE Trans. Pattern Anal. Mach. Intell.* **26** 1105–1111.

[39] LEE, S.-I., LEE, H., ABBEEL, P. and NG, A. Y. (2006). Efficient ℓ_1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence* **21** 401. AAAI Press, Menlo Park, CA.

[40] LOCKHART, R., TAYLOR, J., TIBSHIRANI, R. J. and TIBSHIRANI, R. (2014). A significance test for the Lasso. *Ann. Statist.* **42** 413–468. MR3210970

[41] LUBIN, M. and DUNNING, I. (2015). Computing in operations research using Julia. *INFORMS J. Comput.* **27** 238–248.

[42] MA, S., SONG, X. and HUANG, J. (2007). Supervised group lasso with applications to microarray data analysis. *BMC Bioinformatics* **8** 60.

[43] MARONNA, R., MARTIN, R. D. and YOHAI, V. (2006). *Robust Statistics*. Wiley, Chichester.

[44] MEIER, L., VAN DE GEER, S. and BÜHLMANN, P. (2008). The group lasso for logistic regression. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **70** 53–71.

[45] MENARD, S. (2002). *Applied Logistic Regression Analysis* **106**. Sage, Thousand Oaks, CA.

[46] PREGIBON, D. (1981). Logistic regression diagnostics. *Ann. Statist.* **9** 705–724.

[47] RYAN, T. P. (2009). *Modern Regression Methods*, 2nd ed. Wiley, Hoboken, NJ. MR2459755

[48] SATO, T., TAKANO, Y., MIYASHIRO, R. and YOSHISE, A. (2016). Feature subset selection for logistic regression via mixed integer optimization. *Comput. Optim. Appl.* **64** 865–880. MR3506236

[49] SHAFIEEZADEH-ABADEH, S., MOHAJERIN, P. and KUHN, D. Distributionally robust logistic regression. In *Proceedings of the* 28*th International Conference on Neural Information Processing Systems* (*NIPS'*15), *Montreal*, *Canada*, *December* 07–12, 2015 (C. Cortes, D. D. Lee, M. Sugiyama and R. Garnett, eds.) 1576–1584. MIT Press, Cambridge, MA.

[50] SIMON, N., FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2013). A sparse-group Lasso. *J. Comput. Graph. Statist.* **22** 231–245.

[51] TABACHNICK, B. G., FIDELL, L. S. et al. (2001). *Using Multivariate Statistics*. Allyn and Bacon, Boston, MA.

[52] TIPPING, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **1** 211–244.
[53] YUAN, M. and LIN, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **68** 49–67.
[54] ZHAO, P., ROCHA, G. and YU, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Statist.* **37** 3468–3497.
[55] ZOU, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.* **101** 1418–1429.