This article was downloaded by: [98.217.202.234] On: 13 February 2016, At: 09:26
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

OR Forum—An Algorithmic Approach to Linear Regression

Dimitris Bertsimas, Angela King

To cite this article:
Dimitris Bertsimas, Angela King (2016) OR Forum—An Algorithmic Approach to Linear Regression. Operations Research 64(1):2-16. http://dx.doi.org/10.1287/opre.2015.1436

Full terms and conditions of use: http://pubsonline.informs.org/page/terms-and-conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2015, INFORMS

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

RIGHTSLINK

OPERATIONS RESEARCH
Vol. 64, No. 1, January–February 2016, pp. 2–16
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2015.1436
© 2016 INFORMS

OR Forum—An Algorithmic Approach to Linear Regression

Dimitris Bertsimas, Angela King
Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
{dbertsim@mit.edu, aking10@mit.edu}

Linear regression models are traditionally built through trial and error to balance many competing goals such as predictive power, interpretability, significance, robustness to error in data, and sparsity, among others. This problem lends itself naturally to a mixed integer quadratic optimization (MIQO) approach but has not been modeled this way because of the belief in the statistics community that MIQO is intractable for large scale problems. However, in the last 25 years (1991–2015), algorithmic advances in integer optimization combined with hardware improvements have resulted in an astonishing 450 billion factor speedup in solving mixed integer optimization problems. We present an MIQO-based approach for designing high quality linear regression models that explicitly addresses various competing objectives and demonstrate the effectiveness of our approach on both real and synthetic data sets.

Keywords: integer programming; statistics.
Subject classifications: programming: integer: applications; statistics.
Area of review: OR Forum.
History: Received September 2015; accepted September 2015. Published online in *Articles in Advance* December 30, 2015.

1. Introduction

We consider the linear regression model with response vector $\mathbf{y}_{n\times 1}$, model matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \Re^{n\times p}$, regression coefficients $\boldsymbol{\beta} \in \Re^{p\times 1}$ and errors $\boldsymbol{\epsilon} \in \Re^{n\times 1}$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

The linear regression model is a powerful tool for modeling the relationship between a dependent variable and explanatory variables and is well studied in theory as well as widely applied in practice. However, going from raw data to a high quality linear regression model is a nontrivial task; the modeler must ensure that all modeling assumptions are met while building a parsimonious model that is able to separate signal from noise. The modeler rarely builds a single model. Rather, an iterative process of refinement is applied to produce the best possible model. This task manifests itself as a series of checks during the model building process: is there evidence of multicollinearity? of outliers? Are there too many variables present, or not enough? How well does the model generalize? What about measurement error in the data or missing data? Are the variables significant? Does the resulting model make sense for the application at hand? And so on.

The modeler must balance these competing objectives in the construction of a regression model. In this paper, we propose an algorithmic, optimization-based method for jointly balancing such objectives.

1.1. The Aspirations of the Work

Currently, regression modeling is done in a fairly ad hoc manner. The various properties of a high quality linear regression model are typically built into the model one at a time and through repeated trial and error by the modeler. Hence, there is no guarantee that the final model produced satisfactorily addresses all of them, let alone optimally addresses them. The goal of this work is to design an optimization-based algorithm that simultaneously takes into account these desirable properties and, whenever it is not possible to satisfy all these properties simultaneously, the algorithm provides a guarantee that it is indeed infeasible to do so. The output of such an algorithm is a set of high quality regression models containing as many of the desired properties as possible. As measure of quality we use out-of-sample R^2 and the ability of the model to achieve interpretability, significance, robustness to error in data, and sparsity.

We feel that humans and machines have different strengths and our proposed approach aims to utilize both these strengths. The modeler typically has subject matter expertise; for example, the modeler may know of a particular structure present in the data or can require that certain variables be present in the final model. Whereas humans have intuition and contextual knowledge and understanding, computers have significantly more computational power. Our aspiration in this work is to empower modelers with a methodology that builds models with properties that a human modeler can require based on intuition and expertise.

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

1.2. Current Practice

Fitting regression models has long been viewed as an art, left to the savvy modeler who manages often-competing goals. The result is that two modelers may begin with the same set of data and end with quite different models.

We consulted several widely used regression textbooks (*Regression Analysis by Example* by Chatterjee et al. 2012, *Applied Regression Analysis* by Draper and Smith 1998, *Linear Regression Analysis* by Seber and Lee 2003, and *Applied Linear Regression* by Weisberg 2014) to see how modelers are instructed to approach the difficult task of fitting a linear regression model, and our findings show that although many textbooks discuss these competing objectives individually, most textbooks do not provide guidance to modelers on how to balance these objectives in organizing their search for the best model. For instance, Draper and Smith (1998), Seber and Lee (2003), and Weisberg (2014) each contain a chapter on model selection and discuss topics such as selection criteria, the best subset problem, stepwise methods, shrinkage methods, and computational approaches. Many techniques are offered, but little guidance is provided as to which method a modeler should use, if any, under particular circumstances. The Chatterjee et al. (2012) text also contains a chapter on model selection with a similar set of topics but differs from the other texts in that it also provides a potential strategy for fitting regression models. In our experience, many modelers follow a process similar to what is outlined in Chatterjee et al. (2012). We summarize their suggestions here and henceforth refer to this as "the standard approach":

1. Examine the variables one by one, looking for outliers and making transformations.

2. Construct pairwise scatterplots for each variable, if possible. Examine the correlation matrix and delete redundant variables. Calculate the condition number of the correlation matrix to understand the extent of the effect of multicollinearity.

3. Fit the full ordinary least squares model and delete variables with insignificant t-tests. For the reduced model, examine the residuals for linearity, heteroscedasticity, autocorrelation, and outliers.

4. See if additional variables can be dropped and/or if new variables need to be brought in. Repeat step 3.

5. Check variance inflation factors (VIFs) and residual diagnostics.

6. Validate the fitted model on a test set or use other methods such as cross validation, bootstrapping, etc.

In (Chatterjee et al. 2012, p. 311), the authors are quick to note that the procedure they outline is frequently implemented synchronously rather than entirely sequentially, and that it may be necessary to repeat the steps several times. They qualify their recommended steps by noting that "one important component that we have not included in our outlined steps is the subject matter knowledge of the analyst in the area in which the model is constructed. ... After all is said and done, statistical model building is an art. The techniques that we have described are the tools by which this task can be attempted methodically."

In contrast, our goal is to design an algorithm that eliminates the modeler's tedious task of repeating the model-building steps several times and to produce a high quality set of models.

Table 1. Desirable properties of a linear regression model and how they are incorporated into the model.

Property	Paper section	MIQO model
General sparsity	3.1	Constraint (5d)
Group sparsity	3.2	Constraint (5e)
Limited pairwise multicollinearity	3.2	Constraint (5f)
Nonlinear transformations	3.2	Constraint (5g)
Robustness	3.3	Objective (5a)
Stable to outliers	3.4	Objective (5a)
Modeler expertise	3.5	Constraint (5h)
Statistical significance	3.6	Constraint (5i)
Low global multicollinearity	3.7	Constraint (5i)

1.3. Contribution and Structure of the Paper

In this paper, we propose a mixed integer quadratic optimization (MIQO) 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 MIQO model in §4. Our approach provides the only methodology we are aware of to construct linear regression models that impose statistical properties simultaneously. Using both real and synthetic data, we demonstrate that the approach is generally applicable, is tractable in the sense of providing solutions in realistic timelines, and provides a guarantee of suboptimality because it is based on an MIQO model. Specifically, when the MIQO is infeasible, we obtain a guarantee that imposing distinct statistical properties is simply not feasible.

The paper is structured as follows. We begin in §2 with a brief review of mixed integer optimization and the computational speedups witnessed in the past 25 years. In §3, we introduce and discuss the desirable statistical properties we want the regression model to have. In §4, we develop the MIQO-based algorithm to impose these properties. In §5, we provide evidence of our algorithm's abilities using a wide variety of real and synthetic data sets. We conclude in §6.

2. Mixed Integer Optimization Background

In this section, we present a brief overview of mixed integer optimization (MIO), including the simply astonishing advances it has enjoyed in the last 25 years.

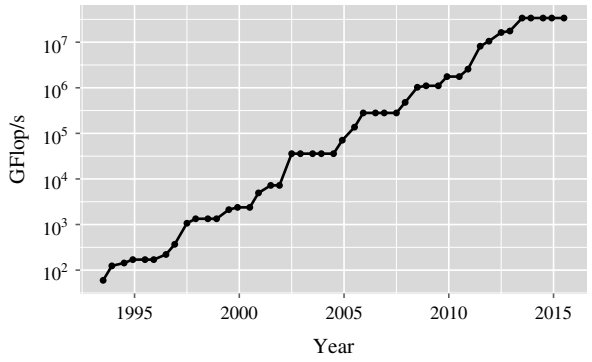
The general form of an MIQO problem is as follows:

$$\begin{aligned} \min \quad & \{\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{a}\} \\ \text{s.t.} \quad & \mathbf{A}\boldsymbol{\alpha} \leqslant \mathbf{b} \\ & \alpha_i \in \{0,1\}, \quad \forall\, i \in \mathcal{I} \\ & \alpha_j \in \mathbb{R}_+, \quad \forall\, j \notin \mathcal{I}, \end{aligned}$$

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

Figure 1. Peak supercomputer speed in GFlop/s (log scale) from 1994 to 2015.



where $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{k\times m}$, $\mathbf{b} \in \mathbb{R}^k$, and $\mathbf{Q} \in \mathbb{R}^{m\times m}$ (positive semidefinite) are the given parameters of the problem; $\mathbb{R}_+$ denotes the nonnegative reals, the symbol $\leqslant$ 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 $\mathcal{I} \subset \{1, \ldots, m\}$. We note that in the MIQO problems, we consider the integer variables are restricted to be binary. For additional background on MIO, see Bertsimas and Weismantel (2005). Subclasses of MIQO problems include convex quadratic optimization problems ($\mathcal{I} = \varnothing$), mixed integer ($\mathbf{Q} = \mathbf{0}_{m\times m}$) and linear optimization problems ($\mathcal{I} = \varnothing, \mathbf{Q} = \mathbf{0}_{m\times m}$). Modern integer optimization solvers such as Gurobi and CPLEX are able to tackle MIQO problems.

In the last 25 years (1991–2015) the computational power of MIO solvers has increased at an astonishing rate. In Bixby (2012), to measure the speedup of MIO solvers, the same set of MIO problems was tested on the same computers using 12 consecutive versions of CPLEX and version-on-version speedups were reported. The versions tested ranged from CPLEX 1.2, released in 1991 to CPLEX 11, released in 2007. Each version released in these years produced a speed improvement on the previous version, leading to a total speedup factor of more than 29,000 between the first and last version tested (see Bixby 2012 and Nemhauser 2013 for details). Gurobi 1.0, an MIO solver that was first released in 2009, was measured to have similar performance to CPLEX 11. Version-on-version speed comparisons of successive Gurobi releases have shown a speedup factor of more than 27 between Gurobi 6.0, released in 2015, and Gurobi 1.0 (Bixby 2012, Nemhauser 2013). The combined machine-independent speedup factor in MIO solvers between 1991 and 2015 is 780,000. This impressive speedup factor is due to incorporating both theoretical and practical advances into MIO solvers. Cutting plane theory, disjunctive programming for branching rules, improved heuristic methods, techniques for preprocessing MIOs, using linear optimization as a black box to be called by MIO solvers, and improved linear optimization methods have all contributed greatly to the speed improvements in MIO solvers (see Bixby 2012).

In addition, the past 20 years have also brought dramatic improvements in hardware. Figure 1 shows the exponentially increasing speed of supercomputers over the past twenty years, measured in billion floating point operations per second, available from Top500.org (2013). The hardware speedup from 1994 to 2015 is approximately $10^{5.75} \sim 570{,}000$. When both hardware and software improvements are considered, the overall speedup is approximately 450 billion! Note that the speedup factors cited here refer to mixed integer linear optimization problems, not MIQO problems. The speedup factors for MIQO problems are similar. 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 25 years, the mindset of MIO as theoretically elegant but practically irrelevant is no longer supported. In this paper, we demonstrate that by using MIQO it is possible incorporate many beneficial statistical properties into the linear regression optimization problem itself. We provide empirical evidence of the success of the method.

3. Desirable Properties of a Linear Regression Model

In this section, we review desirable characteristics of a linear regression model. We discuss our MIQO approach to build these properties into a model and contrast it to other approaches to achieving each property.

3.1. General Sparsity

When the number of potential features is large, we often wish to identify a critical subset that is primarily responsible for producing the response. This leads to more interpretable models and aids prediction accuracy by eliminating noise variables to increase the model's ability to generalize. For this reason, we want to develop linear regression models with a specified number k of nonzero coefficients β. This number k is called the sparsity of the model.

To achieve sparsity, we follow the approach of Bertsimas et al. (2015) and use a combination of continuous and discrete optimization methods to efficiently solve the best subset regression problem (Miller 1990). That is, we will solve the following problem for all values of $k \in \{1, \ldots, p\}$ and return the solution and value of k with the smallest residual sum of squares:

$$\min_{\boldsymbol{\beta}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_0 \leqslant k. \tag{1}$$

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

We will formulate Problem (1) as an MIQO problem and warm-start the MIQO using discrete first order methods as described in Bertsimas et al. (2015). Solving the best subset regression problem as an MIQO provides a solution with a guarantee on its suboptimality even if we terminate the MIQO early. It extends to other objective functions and can accommodate side constraints on the coefficients of the linear regression, and we will heavily take advantage of this to ensure that our linear regression model contains all of the desired properties.

Using optimization to solve the best subset regression problem is not a new approach; the method for best subset regression introduced in Furnival and Wilson (1974) is an MIQO-based method. However, it is only capable of solving the best subset problem accurately for values of $p \leqslant 30$. The MIQO approach outlined in Bertsimas et al. (2015) is significantly more scalable, largely because of the advances in computer hardware, the improvements in MIO solvers, and the specific warm-start techniques developed therein.

Because of the difficulty of scaling algorithms like the approach in in Furnival and Wilson (1974), research on the best subset regression problem in the past few decades has mainly focused on methods that solve a convex approximation of Problem (1). For example, Lasso (Tibshirani 1996, Chen et al. 1998) is a popular model that solves the following problem:

$$\min_{\boldsymbol{\beta}} \left\{ \tfrac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{2}$$

The l_1 penalty on $\boldsymbol{\beta}$ in Problem (2) shrinks the coefficients toward zero and sets many coefficients to be exactly zero, which induces a sparse estimate of $\boldsymbol{\beta}$. Under sufficient regularity conditions, it has been shown that the sparsity pattern of this solution perfectly coincides with the true underlying sparsity pattern (Bühlmann and Van De Geer 2011). However, the regularity conditions required to guarantee this are difficult to verify in practice and are not typically satisfied by highly correlated data—which is a common occurrence in practice.

3.2. Selective Sparsity

We use the term "selective sparsity" to refer to situations where we would like to constrain the joint inclusion of subsets of independent variables. Modeling selective sparsity via MIQO can cover a broad range of settings and we will consider several here: 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 (Eldar and Kutyniok 2012), microarray analysis (Ma et al. 2007), and other applications.

By encoding this structure directly into the MIQO model, we ensure that the resulting solution preserves the group sparsity property. Moreover, MIQO can easily handle overlapping groups—a common phenomenon in microarray data, where some genes may play a role in several functional groups (Jacob et al. 2009). Group sparsity has been highly studied in recent years (for example, see Yuan and Lin 2006, Bach 2008, and Zhao et al. 2009). The most common approach is group Lasso, proposed by Yuan and Lin (2006), and therefore much of the literature focuses on how well the group sparsity property is recovered. With an MIQO approach, a feasible solution guarantees the group sparsity property. To the best of our knowledge, there has not been previous work on group sparsity via MIQO.

Limited Pairwise Multicollinearity. A near-linear relationship between independent variables obfuscates the relationship of each feature to the response and leads to unstable parameter estimates. To avoid these issues and produce interpretable models, a high quality regression model will contain features that are as orthogonal as possible. Thus, we suggest using pairwise correlation as a measurement of multicollinearity and building in selective sparsity by limiting the independent variables in the regression model to those that have relatively low pairwise correlation. This is a standard technique in practice—for example, in their textbook, Tabachnick and Fidell (2001) recommend that independent variables with a pairwise correlation more than 0.70 should not be included in multiple regression analysis.

Other methods of managing multicollinearity include principal components regression (Massy 1965) and partial least squares (Wold et al. 1984), which transform the data to produce new, uncorrelated feature variables. Although these effectively solve the issue of multicollinearity, it may be difficult to interpret the new features and therefore unclear the extent to which the original variables affect the response. Penalized regression, which gives biased estimates but reduces variance, is another common method of attacking the inflated variances that result from multicollinearity. Although this may induce lower variances, the shrinkage induced by these methods does not actually make the data any less correlated, and hence we do not view it as an appropriate tool for encouraging interpretable models.

Detecting Appropriate Nonlinear Transformations. The data may not be collected in the units that are most explanatory of the dependent variable. It may turn out that a nonlinear transformation of an independent variable results in a new variable that can explain the variance in the dependent variable much better than the original measured variable could.

Typically, modelers detect the need for nonlinear transformations through graphical examination and trial and error. We are not aware of any other automated methods of doing this. For a fixed set of nonlinear transformations, MIQO can optimally determine whether to use the original variable or a transformed version of the variable. For any variable

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

j for which nonlinear transformations may be desired, we simply include all the potential transformed versions of the variable in the data set passed to the algorithm. Let the set T_j contain the original variable j and its nonlinear transformations. Then we incorporate selective sparsity by including a constraint in the MIQO model that at most one of the variables from the set T_j can appear in the final model.

3.3. Robustness

Data quality varies widely based on the nature of the data being collected and the collection process. It is very common that the data used in regression models are inaccurate. 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 Ben-Tal et al. 2009 and Bertsimas et al. 2011). For the linear regression problem with data $(\mathbf{y}, \mathbf{X})$, the data associated with the independent variables have error $\Delta\mathbf{X}$ that belong to a given uncertainty set U. For example,

$$U = \{\Delta\mathbf{X} \mid \|\Delta\mathbf{X}\|_{p,q} \leqslant \Gamma\}, \quad \text{where } \|\mathbf{A}\|_{p,q} = \max_{\|\mathbf{z}\|_q = 1} \|\mathbf{A}\mathbf{z}\|_p.$$

The robust least squares problem is then

$$\min_{\boldsymbol{\beta}} \max_{\Delta\mathbf{X} \in U} \tfrac{1}{2}\|\mathbf{y} - (\mathbf{X} + \Delta\mathbf{X})\boldsymbol{\beta}\|_p^p. \tag{3}$$

The key result is as follows.

Theorem 1 (Bertsimas and Fertis 2009, Xu et al. 2009). *Problem* (3) *is equivalent to*

$$\min_{\boldsymbol{\beta}} \left\{ \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_p^p + \Gamma\|\boldsymbol{\beta}\|_q \right\}. \tag{4}$$

For $p = 2$ and $q = 1$, Theorem 1 recovers Lasso. This result demonstrates that penalized regression models like Lasso are actually *robust* models against uncertainty in data. Although Lasso is revered for its ability to induce sparse solutions and much work has been done on the ability of Lasso to recover the true model (see Bühlmann and Van De Geer 2011 for an overview of the conditions under which Lasso identifies the true sparsity pattern), its predictive power is a result of being robust to uncertainty in data. Bertsimas et al. (2015) compare exact subset selection in linear regression models using MIQO and Lasso and report that although the predictive accuracy of MIQO and Lasso are comparable, Lasso produces models that are not particularly sparse. In our algorithm, we regularize our MIQO model as a way to immunize the model from data uncertainty. Because there are many cases in which regularization alone is not able to ensure sparsity (Raskutti et al. 2011, Zhang 2014, Mazumder et al. 2011, Greenshtein et al. 2006, Zhang and Zhang 2012, Shen et al. 2013), we use the regularization approach in addition to the general sparsity approach outlined in §3.1. The robust optimization approach focuses on the worst-case error in the data. The approach is flexible in that it can handle different regularization parameters for different corresponding coefficients. For a characterization of the relation between robustification and regularization, see Bertsimas and Copenhaver (2014).

3.4. Stability Against Outliers

In the ideal modeling scenario, all data are representative of the population from which they are gathered. The presence of outliers can seriously impede the model's generalization ability, so we would like to develop regression models that avoid the effect of outliers. Toward this goal we can use a median regression objective function rather than a least squares objective. The least squares objective is known to produce coefficients that are highly sensitive to outliers. Coefficient sensitivity to outliers is typically quantified using the metric of finite sample breakdown point Donoho and Huber (1983). The least squares objective leads to estimates with a limiting breakdown point of zero (Hampel 1971). The least absolute deviations objective, which minimizes the l_1-norm of the residuals rather than the l_2-norm, also has a breakdown point of zero. However, the least median of squares (LMS) objective, introduced in Rousseeuw (1984), minimizes the median of the l_2-norm of the residuals. The limiting breakdown point of LMS estimators is 50%—the maximum achievable. In Bertsimas and Mazumder (2014), the authors formulate the LMS regression problem using MIQO. Their approach is easily adapted to our setting. To address outliers, we can adopt the LMS objective in place of the least squares objective in Problem (5) while retaining the other constraints and the regularization parameter in the objective.

3.5. Modeler Expertise

In some cases, the modeler has particular expertise with the application at hand. In that case, the modeler might wish to specify that certain independent variables must be included in the final regression model because of a known correlation with the response. This can be incorporated directly into the model building process by adding a constraint to the MIQO model.

3.6. Statistical Significance

Statistical inference relies not just on parameter estimates but also on specifications of uncertainty and confidence regarding those estimates. It is critical when interpreting parameters to have a sense of whether the model is truly detecting an underlying relationship between the variable and the response. The standard way of quantifying this in the scientific literature is through the concept of statistical significance. An independent variable in a regression model is labeled as "statistically significant" if, in the presence of the other variables in the model, the probability α that the observed effect occurred by chance is low, conventionally 5% or less. Modelers typically exclude insignificant variables from regression models because they can only give murky interpretations of their effects on the response.

We would like our algorithm to provide confidence intervals and judge whether a given variable is statistically significant. However, we would also like our methodology to

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

be free of distributional assumptions, to handle high dimensional settings, and to incorporate regularization, as described §3.3. All these properties invalidate the standard least squares assumptions. Our approach, then, is not to compromise these goals but rather to use bootstrapping techniques to generate confidence intervals and test for statistical significance. The bootstrap method was introduced in the seminal paper Efron (1979), and bootstrapped confidence intervals have been shown to be asymptotically more accurate than standard confidence intervals obtained using sample variance and normality assumptions (DiCiccio and Efron 1996).

Coming up with analytical formulae for significance measures and confidence intervals in regularized, potentially high-dimensional settings is challenging and an area of current research (see Javanmard and Montanari 2013 and Lockhart et al. 2014, for example). We prefer our methodology to be flexible and able to handle a variety of objective functions and constraints, and so instead we opt for a bootstrapping approach, which harnesses the power of modern computing.

3.7. Low Global Multicollinearity

We note that it is possible to have multicollinearity without having any high pairwise correlations; see Ryan (2008) for an example where four variables all have pairwise correlation $\leqslant 0.57$ but have a perfect linear relationship. Thus, using a pairwise correlation threshold as a surrogate for eliminating multicollinearity may not catch all cases of multicollinearity.

Global multicollinearity can be measured by checking the condition number of the correlation matrix resulting from the submatrix of included variables. A high condition number indicates a multicollinearity problem. 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 (Chatterjee et al. 2012).

4. Overall Approach

In this section, we describe our overall approach for producing high quality regression models. The method can be applied to any data set that an analyst wishes to model using linear regression. The algorithm is composed of three stages: (1) preprocessing, (2) building and solving the MIQO model, and (3) generating any additional constraints and repeating step (2).

4.1. Stage 1: Preprocessing

The first stage begins with data set preprocessing and parameter setting. The data set 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 10) and ρ, the maximum pairwise correlation that will be allowed between included variables (the default is 0.8). The algorithm then generates the correlation matrix for the training data and identifies variables that are correlated in absolute value beyond ρ and calls this set of pairs of variables $\mathcal{HC}$, for highly correlated variables. The algorithm identifies categorical variables and expresses them as groups of dummy variables. At this point, the modeler can 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 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 $\mathcal{T}_m$. If the modeler believes the data set to contain a significant number of outliers, he can specify at this point to use the median objective function rather than the least squares objective. Finally, the modeler can specify a set $\mathcal{I}$ of variables to be included in the model that capture the modeler's subject expertise. Then the algorithm calculates $k_{\max}$, the maximum possible subset size such that the selective sparsity and modeler expertise constraints are still feasible. This is determined by solving a maximum independent set problem. 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, or stable set, for this graph is a set such that no two vertices are adjacent. The cardinality of this set is exactly equal to the maximum value of k that will result in a feasible MIO model and is the objective value of the following MIO problem:

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

Since the graph contains at least one node, the optimal value $k_{\max}$ is at least one and the algorithm sets the parameter $k_{\max}$ to the objective value and then proceeds to determine a set of Γ values to test. By default, the set is logarithmically spaced between 0 and the value of Γ that would force $\boldsymbol{\beta} = 0$ if the problem were completely unconstrained. This allows a wide variety of robustification parameters to be tested. At this point, all the parameters of the algorithm have been set and the algorithm proceeds to Stage 2.

4.2. Stage 2: The MIQO model

The algorithm solves the following MIQO model for each value of k from 1 to $k_{\max}$ and each value of Γ using the training data $\mathbf{y}$ and $\mathbf{X}$.

$$\min_{\boldsymbol{\beta}, \mathbf{z}} \ \left\{ \tfrac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \Gamma \|\boldsymbol{\beta}\|_1 \right\}, \tag{5a}$$

$$\text{s.t.} \quad z_l \in \{0,1\}, \quad l = 1, \ldots, p, \tag{5b}$$

$$-\mathcal{M} z_l \leqslant \beta_l \leqslant \mathcal{M} z_l, \quad l = 1, \ldots, p, \tag{5c}$$

RIGHTSLINK

$$\sum_{l=1}^{p} z_l \leqslant k, \tag{5d}$$

$$z_1 = \cdots = z_l(1, \ldots, l) \in \mathcal{GS}_m, \quad \forall m, \tag{5e}$$

$$z_i + z_j \leqslant 1 \quad \forall (i, j) \in \mathcal{HC}, \tag{5f}$$

$$\sum_{i \in \mathcal{T}_m} z_i \leqslant 1 \quad \forall m, \tag{5g}$$

$$z_l = 1 \quad \forall l \in \mathcal{I}, \tag{5h}$$

$$\sum_{l \in \mathcal{S}_i} z_l \leqslant |\mathcal{S}_i| - 1 \quad \forall \mathcal{S}_1, \ldots, S_j. \tag{5i}$$

In the objective function (5a), the robustification parameter Γ immunizes the resulting model against uncertainty in the data; see Equation (4). In constraint (5b), a binary indicator variable z_l is introduced for every β_l in the model. For a large enough constant $\mathcal{M}$, the constraint (5c) ensures that β_l will be nonzero only if $z_l = 1$. The parameter $\mathcal{M}$ can be estimated from data (see Bertsimas et al. 2015 for details). The constraint (5d) limits the number of total variables that will be included in the model. This ensures general sparsity of the resulting model. The constraints in (5e)–(5g) are selective sparsity constraints. For the mth set of variables with a group sparsity structure, the set of constraints defined in (5e) ensures that the variables in $\mathcal{GS}_m$ are either all zero or all nonzero. The set of constraints in (5f) ensures that the resulting model is free from extreme pairwise multicollinearity. The set $\mathcal{T}_m$ refers to the mth variable that was flagged as a candidate for transformation and all of its possible nonlinear transformations. The set of constraints (5g) ensure 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 (5h) will be included in the model and will ensure that each of the specified independent variables appears in the final model. (5i) is a set of constraints to exclude particular solutions $\mathcal{S}_i$, such as those with high global multicollinearity or containing variables that are statistically insignificant. $\mathcal{S}_i$ is the set of indices corresponding to nonzero $\boldsymbol{\beta}$ value in the ith solution. The initial MIQO model will not contain line (5i); these constraints will be generated in Stage 3, if necessary.

The output of the MIQO model is a set of variables $\boldsymbol{\beta}^*$ and $\mathbf{z}^*$. We measure and record the out-of-sample R^2 on the validation set using this $\boldsymbol{\beta}^*$. Once the MIQO model is run for all potential values of k and Γ, the algorithm chooses the three sets of $\boldsymbol{\beta}$ with the highest R^2 on the validation set as the top three regression models and proceeds to Stage 3.

4.3. 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 and calculates the condition number of the model. If a set $\mathcal{S}_i$ produces undesirable results—a condition number higher than desired or a model with insignificant variables—the algorithm generates the MIQO constraint (5i) 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 $\sum_{l \in \mathcal{S}_i} z_l \leqslant |S_i| - 1$. For example, to exclude set $\mathcal{S}_1 = \{1, 4, 7\}$, we can insert the constraint $z_1 + z_4 + z_7 \leqslant 2$ into Problem (5) 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 that is acceptable in the model as well as the number of iterations permitted between Stage 2 and Stage 3. The defaults are 30 and 3, respectively. In our experience, if a linear 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. Confidence histograms and diagnostic plots can also be generated.

4.4. Contrast with the Standard Approach

In many ways, our algorithm simply automates several of the steps outlined in the standard approach. Moreover, both our algorithm as well as the standard approach validate models out of sample rather than relying on in-sample criteria. This ensures that the model selected does not overfit the training data. However, we highlight a few key differences.

1. Our algorithm does not have to choose which model properties to favor by performing the steps in a certain order; since it is based on optimization, these properties can be addressed jointly rather than sequentially. For example, rather than noticing pairwise multicollinearity and preemptively deleting one variable, our MIQO model simply chooses which variable is best to delete in the course of the optimization.

2. Our algorithm is capable of handling data sets with more variables than a modeler can address manually. The steps suggested in Chatterjee et al. (2012) become difficult when p is large, and the modeler must often resort to a computational method for variable selection prior to performing the rest of the steps.

3. Our algorithm is capable of returning a set of high quality models rather than focusing on refining a single good model.

As an example, we illustrate our algorithm's performance on two data sets and compare it to a model that a modeler might develop using these data.

4.5. Example 1

We compare and contrast our algorithm with the standard approach using the Croq'Pain data set from Bertsimas and Freund (2004).

The data set originally comes from Croq'Pain, a French "restaurant rapide," and contains data on sixty Croq'Pain stores. For each store, the data set provides information on the store and the surrounding area. There are a total of 16 variables provided per store (see Table 2 for details).

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

Table 2. Variables in the Croq'Pain data set.

Variable	Description
EARN	Operating earnings in \$1,000s
SIZE	Total area inside store
EMPL	Number of employees as of Dec. 31, 1994
P15	Number of 15- to 24-year-olds in a 3 km radius
P25	Number of 25- to 34-year-olds in a 3 km radius
P35	Number of 35- to 44-year-olds in a 3 km radius
P45	Number of 45- to 54-year-olds in a 3 km radius
P55	Number of people age 55+ in a 3 km radius
TOTAL	Total population in a 3 km radius
INC	Average income in town/neighborhood surrounding site
COMP	Number of competitors in 1 km radius
NCOMP	Number of restaurants that do not compete with Croq'Pain in a 1 km radius
NREST	Number of non-restaurant businesses in a 1 km radius
PRICE	Monthly rent per square meter of retail properties in the same locale
CLI	Cost of living index
K	Invested capital

The case described in Bertsimas and Freund (2004) asks the student to use these data to build a regression model to help Croq'Pain decide whether to open a new store. The decision will be based on the store's performance ratio, which is measured as the ratio of operating earnings to invested capital. The goal is to build a high quality regression model with performance ratio as the dependent variable, and the first step of this—the fitted model with all independent variables included—is given in Bertsimas and Freund (2004). Recall that we measure predictive quality using out of sample R^2.

The Standard Approach. The model with all 14 independent variables has an R^2 value of 0.867. Five of the 14 variables are significant at the 0.05 level, and it seems that some of the coefficient estimates may take the opposite signs from what is expected. For example, the coefficients for number of employees and for total surrounding population are both negative.

A quick look at the correlation matrix shows that there are a number of independent variables which are highly correlated: for example, P35 and TOTAL have a correlation coefficient of 0.96. However, the 14×14 matrix is unwieldy to work with manually. Instead of trying to eliminate correlated variables first, we begin to refine this model by removing variables that are insignificant at the 0.05 level, starting with those with the lowest t-value. Removing variables one at a time according to this method until all variables left are significant results in a new model with only five independent variables (SIZE, P15, INC, NREST, and PRICE) and a training set R^2 value of 0.856. At this point, the number of independent variables is low enough to investigate the correlation matrix manually. None of the remaining five independent variables has correlation over 0.18 in magnitude, so we feel assured that multicollinearity is not a problem in this reduced model. We "sign-check" each of the remaining five independent variables and validate that the signs agree with our intuition. We move on to residual diagnostics and check for normality of the residuals by plotting a histogram and for heteroscedasticity by plotting each of the independent variables against the residuals. There is no evidence of nonnormality or of heteroscedasticity. Therefore, we use this model as the final model.

MIQO-Based Approach. In the original case in Bertsimas and Freund (2004), the students are first instructed to train their model using the entire data set. The second part of the case asks them to rebuild using the first 50 data points to train the model and the last 10 to validate. Because our MIQO-based approach requires a training set and a validation set, we adopt the second option.

We run our MIQO-based algorithm on the data set using the default settings: 0.8 as the maximum pairwise correlation and 10 potential values of Γ. It takes less than one minute to run and returns a model with five independent variables: SIZE, P15, INC, NREST, and PRICE; exactly the same five we chose via the standard approach. The first four variables are significant at the 0.001 level, the last at the 0.01 level. The model has an out-of-sample validation set R^2 value of 0.80.

With the Croq'Pain data, the MIQO-based approach and the standard approach produced essentially the same model. In cases like these, we feel the main advantage of the MIQO-based approach is the amount of time saved from iterating through potential models. Although the computational time executing the MIQO-based approach is longer, the total time spent model building is far shorter. In other cases, the standard approach may not lead to as clear of a path to a high quality solution, or the data set may contain enough variables to render it intractable for a human modeler. It is these cases for which the MIQO-based approach is not simply a time saver but a strong improvement over existing tools. The next example illustrates this case.

4.6. Example 2

We consider the Ames Housing Data set (DeCock 2011). The data set originally comes from the Ames City Assessor's Office and contains data on property sales in Ames, Iowa, between 2006 and 2010. The variables include discrete, continuous, nominal, and ordinal variables that describe the quality and quantity of physical attributes of each property sold. The physical attributes measured include building type and style, square footage and lot details, quality and materials of the property's interior and exterior, and many more. The data set was curated for use as a final group project in a semester-long regression class and is available along with full details in DeCock (2011). The prepared data set contains 2,930 observations and 80 variables. After expanding categorical variables into dummy variables and removing outliers and missing values, the final number of observations and variables is 2,271 and 315, respectively. The potential project described in DeCock (2011) asks the

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

student to use these data to build a regression model to predict housing sale prices.

The Standard Approach. This data set is large and complex enough that there is no single clear best model. Indeed, this is the motivation in DeCock (2011) behind assigning this data set as a final course project; the richness of the data leads to fruitful discussion of students' different approaches. DeCock (2011) mentions that by using only the categorical variable for neighborhood and the two continuous variables that together make up the property's total square footage leads to a model that explains 80% of the variability. At the other end of the spectrum, the author also admits to spending a fair amount of time constructing a 36-variable model (using some variables he created through recoding and interactions) that explains 92% of the variation in sales. DeCock (2011) does not give further details of the model. Although this may be overly complicated, it illustrates the challenge of building a high quality regression model using the standard approach.

MIQO-Based Approach. After removing missing values and splitting the data set into training, validation, and test sets, we ran our MIQO-based algorithm on the data set using the default settings: 0.8 as the maximum pairwise correlation and 10 potential values of Γ.

The best model generated contained 20 independent variables, all of which were significant at the 0.05 level, and had a test set R^2 value of 0.920. This is competitive with the predictive power of the more complicated 36-variable model constructed by DeCock (2011) while being more interpretable and still retaining statistically significant variables. The best MIQO model contained variables such as the overall quality and condition of the property, whether the property is identified as being in a particular neighborhood, the number of half bathrooms, whether the foundation of the home was constructed from stone or not, the year the garage was built, various measurements of square footage, and the type of electrical system.

5. Computational Experiments

In this section, we illustrate our algorithm's capabilities by demonstrating its performance on a wide variety of data sets. We include data sets that are real as well as synthetic, that are from the classical overdetermined regime with $n > p$ as well as from the undetermined high-dimensional regime with $n < p$, and that contain various different structures and built-in properties. Our goal is to demonstrate that all of the desirable characteristics outlined in §3 can be achieved with MIQO in practical settings.

We begin by examining basic data sets, where all variables are continuous and there is no special structure. In such data sets, the main properties we would like to ensure are interpretability (via general sparsity and limited pairwise multicollinearity constraints) and robustness (via a regularization parameter in the objective function). We consider synthetic examples to highlight the algorithm's performance on these properties individually and real data sets in which we look at desirable properties jointly. In each case, we compare our algorithm's performance to Lasso because Lasso is designed to give interpretability and robustness.

We then provide results for data sets with additional features: data sets with the group sparsity property, with variables that need a nonlinear transformation, with outliers, and so on. We again compare our results to Lasso and compare them also to the published approach taken by the modeler, if available, or to specific algorithms designed for the setting at hand (e.g., group Lasso for the group sparsity case).

Synthetic Data. We generated data such that $\mathbf{x}_i \sim \mathrm{N}(\mathbf{0}, \Sigma)$, $i = 1, \ldots, 2n$ are independent realizations from a p-dimensional multivariate normal distribution with mean zero and covariance matrix $\Sigma := (\sigma_{ij})$. The data were randomly split 50%/25%/25% into training, validation, and test set, respectively. 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: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$. We denote the number of nonzeros in $\boldsymbol{\beta}$ by k. The choice of $\mathbf{X}, \boldsymbol{\beta}, \sigma$ determines the signal-to-noise ratio (SNR) of the problem, which is defined as

$$\mathrm{SNR} = \frac{\mathrm{var}(\mathbf{x}'\boldsymbol{\beta})}{\sigma^2}.$$

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 $k = 10$ and $\beta_i = 1$ for $i \in \{1, \ldots, p\}$ such that $i \bmod p/k = 0$ to generate k equally spaced values.

Real Data. We tested our algorithm on twelve publicly available datasets. The datasets White Wine Quality, Red Wine Quality, Yacht Hydrodynamics, and CPU were obtained from the University of California Irvine Machine Learning Repository (Bache and Lichman 2014). We obtained the data sets Elevator, Pyrimidines, and Compact from a data repository at the University of Porto (Torgo 2014). We obtained the data sets LPGA 2008, LPGA 2009, and Airline Costs from a data repository at the University of Florida (Winner 2014). We obtained the Diabetes data set from the `lars` package within `R`. The HIV data set comes from the study Rhee et al. (2006) and is available at Hastie (2015).

Computational Specifications. All computational tests were performed on a Dell Precision T7600 computer with an Intel Xeon E52687W (3.1 GHz) processor, 16 cores, and 128 GB of RAM. We used `Gurobi 6.0.0` (Gurobi Inc. 2014) as the optimization solver, and implemented the algorithm in `Julia 0.3.3` (Bezanson et al. 2012), a technical computing language. We used `JuMP 0.7.0` (Lubin and Dunning 2015), 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. We used the `grplasso` package 0.4–4 in `R` (R Core Team 2014) to compute group Lasso solutions.

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

5.1. Basic Structure

Our main goals are to achieve interpretability and robustness while retaining predictive power. To judge how well our algorithm achieves interpretability, we will report on the size k of the subset chosen, the maximum pairwise correlation, and the condition number of the final model. Although our algorithm returns the top three models, we only present results for the top model for brevity. To judge robustness and predictive power, we will report on the Γ chosen by the algorithm on the validation set and the test set R^2 value. We will compare these results to the size k of the subset chosen by Lasso, the maximum pairwise correlation in the Lasso model, and the test set R^2 value in the Lasso model. We selected the best regularization parameter for Lasso by its performance in the validation set. For the synthetic data sets, we also report the number of true positives achieved by each algorithm.

We aim to return solutions in practical amounts of time, so we imposed a time limit on each optimization problem solved: 20 seconds in the $n > p$ case and 40 seconds in the $n < p$ case. Often optimality is reached before the time limit. Note that for each data set, $K_{\max} \times$ (# of values of Γ tested) $\times$ (# of iterations of Stage 3) MIQO problems are solved.

We present results in Tables 3–8 for synthetic data sets for the default parameters of the algorithm: 10 values of Γ tested and 0.8 as the maximum pairwise correlation allowed. Each experiment corresponds to two rows in a table. The top row presents average results over five trials of the same experiment and the bottom row presents the standard error. We use the following notation: SNR = signal-to-noise ratio, K^* = value of k chosen by the algorithm, TP = number of true nonzero variables identified by the algorithm, MaxCor = the maximum pairwise correlation present in the final model, and Cond = condition number. Time for the MIQO algorithm is presented in hours 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.

Tables 3 and 4 show results for data sets designed to illustrate general sparsity, for the $n > p$ and $n < p$ case, respectively. Here we observe that the MIQO algorithm consistently identifies the true nonzero variables and does not bring more than one to two additional noise variables into the model. In contrast, Lasso does correctly identify the true nonzero variables but brings ≈ 24 noise variables into the model in the $n > p$ case and ≈ 45 noise variables into the model in the $n < p$ case. The MIQO models and Lasso models perform similarly in terms of predictive power (out of sample R^2).

Tables 5 and 6 show results for data sets designed to illustrate pairwise multicollinearity for the $n > p$ and $n < p$ case, respectively. Again in these cases, the MIQO models and Lasso models perform similarly in terms of predictive power. However, the final Lasso models contain very high pairwise collinearity and condition numbers that indicate severe multicollinearity issues. On the other hand, the MIQO algorithm returns models that generally have half or less of the maximum pairwise collinearity as the corresponding Lasso model, and the condition numbers do not show evidence of severe multicollinearity.

Tables 7 and 8 show results for data sets designed to illustrate robustness for the $n > p$ and $n < p$ case, respectively. As described in §3.3, Lasso is designed to be robust to error in data. Indeed, in both the $n > p$ and $n < p$ case, Lasso and the MIQO algorithm achieve similar predictive power. The maximum pairwise collinearity and condition numbers of the MIQO-based models are lower.

We present results on real data in Table 9. All optimization problems were solved to optimality except for the Diabetes data set and HIV data set, where a time limit of 20 seconds per optimization problem solved was enforced. Again, for each data set, $K_{\max} \times$ (# of values of Γ tested) $\times$ (# of iterations of Stage 3) MIQO problems are solved. Note that n here indicates the size of the training data set—the original data set has $2n$ observations.

Our algorithm achieves similar predictive performance to Lasso but is significantly more interpretable, choosing fewer variables in general and successfully limiting the degree of multicollinearity present in the final model.

We notice that the Pyrimidines data set has significantly lower predictive power than Lasso. This is the price of insisting on interpretability, despite a relatively low ratio of observations to variables. We demonstrate the algorithm's performance on this data set when the maximum correlation threshold is set to 1 (i.e., no limit) and record the performance in Table 10.

In these cases, it is up to the analyst to judge which model is preferable; one with better predictive performance or one with coefficients that are more interpretable. The benefit of using our algorithm is that it is simple for an analyst to tweak the parameters and quickly understand the tradeoffs.

5.2. Special Structure

Nonlinear Transformations. We investigate our algorithm's capability to identify when a nonlinear transformation of an independent variable may be useful. For this task, we used the Concrete Compressive Strength data set from Yeh (1998) available in the UCI Machine Learning Repository Bache and Lichman (2014). The data set contains 8 independent variables and 1,030 observations. As before, we randomly split the data set into a training set (50%), validation set (25%), and test set (25%).

The independent variable is the compressive strength of concrete, and the dependent variables are the ingredients as well as the age of the concrete (see Table 11 for details). The practical goal in civil engineering is to design a concrete mixture that will have high compressive strength. However, concrete compressive strength is known to be a highly nonlinear function of its age and ingredients.

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

Table 3. Sparsity: $n = 500$, $p = 100$, $\rho = 0$, $\mathbf{\Delta X} = \mathbf{0}$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
6.32	0.014	10.6	10	0.716	0.119	1.61	0.448	34.8	10	0.701	0.148	2.782
	0.010	0.358	0	0.007	0.007	0.02	0.011	3.51	0	0.007	0.010	0.127
3.16	0.011	10.6	10	0.909	0.119	1.27	0.439	34.4	10	0.904	0.148	2.805
	0.010	0.358	0	0.003	0.007	0.29	0.011	3.72	0	0.002	0.010	0.146
1.58	0.011	10	10	0.975	0.117	1.58	0.304	34.6	10	0.974	0.160	2.797
	0.009	0	0	0.001	0.007	0.04	0.011	4.40	0	0.001	0.016	0.194

Table 4. Sparsity: $n = 100$, $p = 500$, $\rho = 0$, $\mathbf{\Delta X} = \mathbf{0}$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
10.54	0.107	10.2	10	0.991	0.249	2.664	1.00	55	10	0.982	0.323	139
0	0.028	0.179	0	0.000	0.026	0.195	0.08	7.33	0	0.001	0.006	97
6.32	0.041	10	10	0.976	0.231	2.793	1.18	56	10	0.952	0.323	1,472
0	0.023	0	0	0.001	0.018	0.217	0.00	8.78	0	0.003	0.006	1,290
3.16	0.076	11	10	0.896	0.216	3.348	1.64	58.2	10	0.813	0.343	5,215
0	0.027	0.283	0	0.006	0.012	0.192	0.42	9.10	0	0.012	0.014	4,634

Table 5. Pairwise multicollinearity: $n = 500$, $p = 100$, true $K = 10$, $\rho = 0.9$, $\mathbf{\Delta X} = \mathbf{0}$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
8.73	0.02	10.00	10.00	0.99	0.40	4.15	0.30	34.40	10.00	0.99	0.91	126.28
	0.01	0.00	0.00	0.00	0.01	0.17	0.02	2.65	0.00	0.00	0.00	13.15
4.37	0.02	10.40	10.00	0.95	0.47	5.65	0.34	37.20	10.00	0.94	0.91	146.36
	0.02	0.36	0.00	0.00	0.07	1.25	0.04	3.66	0.00	0.00	0.00	20.83
2.18	0.03	11.40	9.60	0.81	0.63	7.92	0.63	36.60	10.00	0.81	0.91	142.17
	0.02	0.54	0.22	0.01	0.08	2.25	0.15	3.37	0.00	0.01	0.00	18.89

Table 6. Pairwise multicollinearity: $n = 100$, $p = 500$, true $K = 10$, $\rho = 0.8$, $\mathbf{\Delta X} = \mathbf{0}$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
10.54	0.090	10.4	10	0.990	0.331	5.58	1.99	56.2	10	0.979	0.850	119.8
0	0.029	0.358	0	0.001	0.089	1.48	0.517	2.92	0	0.004	0.005	8.2
6.32	0.049	10.4	10	0.976	0.436	6.11	2.12	57	10	0.941	0.846	122.19
0	0.020	0.219	0	0.003	0.118	1.24	0.395	2.65	0	0.010	0.005	7.47
3.16	0.037	12.4	8.8	0.835	0.433	4.38	2.11	61.2	9.8	0.768	0.846	245.4
0	0.011	0.219	0.72	0.041	0.099	0.51	0.514	3.70	0.179	0.029	0.005	117.9

Table 7. Robustness: $n = 500$, $p = 100$, true $K = 10$, $\rho = 0$, $\mathbf{\Delta}X \sim \text{Uniform}(0, 2)$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
6.32	0.011	10	10	0.975	0.117	1.58	0.448	34.6	10	0.974	0.160	2.797
	0.009	0.000	0	0.001	0.007	0.04	0.011	4.40	0	0.001	0.016	0.194
3.16	0.011	10.6	10	0.909	0.119	1.27	0.439	34.4	10	0.904	0.148	2.805
	0.010	0.358	0	0.003	0.007	0.29	0.011	3.72	0	0.002	0.010	0.146
1.58	0.014	10.6	10	0.716	0.119	1.61	0.304	34.8	10	0.701	0.148	2.782
	0.010	0.358	0	0.007	0.007	0.02	0.011	3.51	0	0.007	0.010	0.127

Table 8. Robustness: $n = 100$, $p = 500$, true $K = 10$, $\rho = 0$, $\mathbf{\Delta X} \sim \text{Uniform}(0, 1)$.

SNR	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
10.54	0.065	10.6	9.6	0.880	0.282	2.777	1.173	53.8	10	0.856	0.376	53.8
0	0.036	0.607	0.358	0.034	0.021	0.131	0.000	4.66	0	0.013	0.018	22.5
6.32	0.044	10	9.4	0.828	0.246	2.716	1.643	53.4	10	0.829	0.357	107.3
0	0.025	0.632	0.358	0.033	0.017	0.268	0.419	7.69	0	0.029	0.014	75.3
3.16	0.038	11	9.4	0.769	0.262	2.475	1.876	61.8	10	0.705	0.338	763.0
0	0.025	0.85	0.358	0.030	0.027	0.585	0.420	10.27	0	0.035	0.010	546.8

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

Table 9. Results for basic structure real data sets.

Data set	n	p	MIQO K^*	R^2	MaxCor	Lasso K^*	R^2	MaxCor
CPU	105	6	5	0.869	0.716	6	0.861	0.716
Yacht	154	6	1	0.600	NA*	1	0.602	NA*
White quality	2,499	11	10	0.270	0.619	9	0.280	0.828
Red quality	800	11	6	0.384	0.40	7	0.386	0.69
Compact	4,096	21	15	0.717	0.733	21	0.725	0.942
Elevator	8,280	18	10	0.808	0.678	15	0.809	0.999
Pyrimidines	37	26	15	0.175	0.781	20	0.367	0.928
LPGA 2008	78	6	2	0.877	0.02	3	0.873	0.234
LPGA 2009	73	11	7	0.814	0.784	10	0.807	0.943
Airline costs	15	9	2	0.672	0.501	9	0.390	0.973
Diabetes	221	64	4	0.334	0.423	14	0.381	0.672
HIV	528	98	11	0.945	0.662	39	0.944	0.760

*Note that both the MIQO and Lasso algorithms choose only one independent variable for the Yacht Hydrodynamics data set; hence, there is no maximum pairwise correlation in this case.

Table 10. Results when the maximum correlation threshold is set to 1.

Data set	n	p	MIQO K^*	R^2	MaxCor	Lasso K^*	R^2	MaxCor
Pyrimidines	37	26	18	0.375	0.870	20	0.367	0.928

On the original data, both our algorithm and Lasso chose to use all covariates and produced a test set R^2 of 0.609. We then reran our algorithm with an extended data set, which contained each of the original columns x as well as three transformed versions of each column: x^2, $\sqrt{x}$, and $\log(x)$. For the variables that take zero values (blast furnace slag, fly ash, and superplasticizer), we adjusted the log transformation to be $\log(x+0.00001)$. We included Constraint (5g) in the optimization model to ensure that for each column x, at most one of x, x^2, $\sqrt{x}$, and $\log(x)$ appeared in the final model.

As expected, the inclusion of transformed covariates significantly improved upon the models created with just the original variables. The MIQO algorithm selected six covariates to appear in the top model. The six covariates chosen were blast furnace slag, water, log(fly ash), log(super plasticizer), log(day), and cement$^{1/2}$. Each covariate was significant at the $\alpha = 0.001$ level and test set R^2 was 0.823, a significant improvement over the original test set R^2 of 0.609.

We also tested Lasso on the data set that included the nonlinear transformations. Lasso selected a model with 12 covariates that resulted in a test set R^2 of 0.834. In addition to the first five covariates chosen by our algorithm, Lasso also selected cement^2, super plasticizer^2, day^2, log(cement), log(coarse aggregate), log(fine aggregate), and $\sqrt{\text{cement}}$. In our opinion, the minor increase in test set R^2 does not warrant using a significantly less interpretable model.

Although the number of variables went from 8 to 32 when we included nonlinear transformations, the MIQO algorithm took the same amount of time (roughly 1–1.5 minutes) to execute in both cases. By imposing limiting constraints on transformations and pairwise correlation, the feasible space is not significantly enlarged by including nonlinear transformations.

Table 11. Independent variables in the concrete compressive strength data set.

Variable	Units
Concrete compressive strength	MPa
Cement	kg/m^3
Blast furnace slag	kg/m^3
Fly ash	kg/m^3
Water	kg/m^3
Superplasticizer	kg/m^3
Coarse aggregate	kg/m^3
Fine aggregate	kg/m^3
Age	Day

Group Sparsity. We demonstrate our results in a group sparsity setting using the Energy Efficiency data set from Tsanas and Xifara (2012) available on the UCI Machine Learning Repository (Bache and Lichman 2014). The data set has 768 observations of six continuous independent variables and two categorical independent variables. The independent variables describe building properties (see Table 12 for details). There are two dependent variables available: heating

Table 12. Independent variables in the energy efficiency data set.

Variable	Type
Relative compactness	Continuous
Surface area	Continuous
Wall area	Continuous
Roof area	Continuous
Overall height	Continuous
Orientation	Categorical; 4 levels
Glazing area	Continuous
Glazing area distribution	Categorical; 6 levels

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

load and cooling load. We test our method on both dependent variables.

The binary expansion of the categorical variable orientation into three new binary variables and of glazing area distribution into five new binary variables meant that the data set passed to the algorithm contained 14 independent variables. When we ran the algorithm, the best model contained three variables: wall area, overall height, and glazing area. We found identical results when predicting cooling load. Our algorithm chose not to use either of the categorical variables provided. The top heating load model had test set R^2 of ≈ 0.88 and the top cooling load model had test set R^2 values of ≈ 0.85.

In Tsanas and Xifara (2012), the original study of this data set, the authors found that wall area, roof area, and relative compactness were the variables that appear mostly associated with heating load and cooling load, although all variables appear in their model. Using the Random Forest method, they also found importance scores for each variable and found that glazing area was the most important variable even though it is not the most correlated with either output variable. However, from an engineering perspective, it can be intuitively understood that the glazing area is of paramount significance.

We find it notable that our algorithm did not identify the same three variables as most critical for predicting the responses and could not have: because of the correlation of -0.86 between roof area and relative compactness, these two variables could not have both been in our algorithm's final model. However, glazing area, which the authors point out as having paramount significance, is in all three of our top models for both response variables.

We also tested group Lasso. The group Lasso models for predicting the two response variables each chose to use 13 of the 14 variables, including both categorical variables. The only variable excluded was surface area. The heating load model had a test set R^2 of ≈ 0.91 and the cooling load model had a test set R^2 value of ≈ 0.86.

5.3. Combined Example

In the previous sections we have demonstrated how our algorithm can handle a wide variety of individual situations: detecting sparsity, limiting pairwise correlation, identifying nonlinear transformations, and others. In this section, we will show the full force of our algorithm: to identify all these 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 data set. We test this example on the high-dimensional case where $n = 100$ and $p = 1{,}000$.

We generated a synthetic data matrix $\mathbf{X}$ for $n = 100$, $p = 500$ according to the process outlined in §5.1, using a value of $\rho = 0.8$ to ensure that there is high pairwise multicollinearity present between some columns of $\mathbf{X}$. 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 1,000. As before, we consider $k = 10$. However, we generated $\beta_i = 1$ so that seven positive values occurred in the original 500 columns and three were located in the 500 transformed columns. The response $\mathbf{y}$ was generated as before as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. To test our robustness to error in data, we generated a matrix $\Delta\mathbf{X} \sim \text{Unif}(0, f)$ and considered $\mathbf{X} + \Delta\mathbf{X}$ for various values of f. 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

Table 13. Results for the combined example.

ϵ	ΔX	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0	0.026	10.4	10	0.981	0.437	4.654	1.14	46.6	10	0.969	0.836	118.1
		0.020	0.219	0	0.001	0.020	0.382	0.17	4.15	0	0.004	0.007	17.4
0.5	1	0.000	11.2	10	0.913	0.556	6.995	1.34	65.8	10	0.854	0.798	424.0
		0.000	0.522	0	0.013	0.073	1.422	0.22	6.97	0	0.017	0.006	177.5
0.5	2	0.030	11.0	9	0.742	0.501	5.291	1.88	69	9.2	0.598	0.708	8,147.1
		0.027	0.490	0.283	0.030	0.061	0.508	0.42	8.54	0.179	0.045	0.006	6,993.5
1	0	0.026	11.2	10	0.931	0.468	5.322	1.08	45.6	10	0.878	0.836	113.9
		0.022	0.522	0	0.007	0.018	0.531	0.04	3.99	0	0.016	0.007	18.2
1	1	0.041	10.4	10	0.878	0.478	4.998	1.61	69.2	10	0.759	0.796	362.8
		0.036	0.219	0	0.016	0.059	0.696	0.43	4.92	0	0.033	0.006	96.4
1	2	0.099	9.8	7.6	0.573	0.436	4.224	1.64	72.4	8.6	0.503	0.702	573.8
		0.041	0.867	0.219	0.042	0.061	0.576	0.42	5.89	0.358	0.064	0.006	228.0
2	0	0.090	10	8.8	0.720	0.451	4.687	1.35	39.4	8.6	0.599	0.836	74.0
		0.045	0.283	0.335	0.046	0.025	0.289	0.23	4.01	0.456	0.055	0.007	9.76
2	1	0.116	9.4	8.2	0.614	0.426	4.262	2.05	53.4	7.8	0.509	0.782	113.5
		0.037	0.358	0.593	0.078	0.025	0.137	0.39	4.41	0.955	0.067	0.010	20.7
2	2	0.032	8.2	4	0.245	0.403	3.506	1.46	55.8	5.8	0.368	0.680	8,141.4
		0.017	0.657	0.980	0.129	0.074	0.720	0.22	10.1	0.522	0.061	0.011	7,236.0

RIGHTSLINK

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

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 13 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.

6. Conclusions

In this paper, we have leveraged the power of MIQO and proposed an approach for incorporating a variety of desired properties into a linear regression model. Our approach provides the only methodology we are aware of to construct models that impose statistical properties simultaneously. This results in a generally applicable, unified framework for addressing all aspects of the model-building process. Using both real and synthetic data, we demonstrate that the approach produces high quality linear regression models in realistic timelines.

Acknowledgments

The authors would like to thank the area editor Edieal Pinker and the two reviewers of the paper for helpful suggestions that improved the paper. Research supported in part by the MIT-Accenture alliance.

References

Bach FR (2008) Consistency of the group Lasso and multiple kernel learning. *J. Machine Learn. Res.* 9:1179–1225.

Bache K, Lichman M (2014) UCI machine learning repository. Accessed August 20, 2014, http://archive.ics.uci.edu/ml.

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust Optimization* (Princeton University Press, Princeton, NJ).

Bertsimas D, Copenhaver M (2014) Characterization of the equivalence of robustification and regularization in linear, median, and matrix regression. Submitting for publication.

Bertsimas D, Fertis A (2009) On the equivalence of robust optimization and regularization in statistics. Technical report, http://www.mit.edu/~dbertsim/papers.html.

Bertsimas D, Freund R (2004) *Data, Models, and Decisions: The Fundamentals of Management Science* (Dynamic Ideas Press, Belmont, MA).

Bertsimas D, Mazumder R (2014) Least quantile regression via modern optimization. *Ann. Statist.* 42(6):2494–2525.

Bertsimas D, Weismantel R (2005) *Optimization Over Integers*, Vol. 13 (Dynamic Ideas Press, Belmont, MA).

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev.* 53(3):464–501.

Bertsimas D, King A, Mazumder R (2015) Best subset selection via a modern optimization lens. *Ann. Statist.* Forthcoming.

Bezanson J, Karpinski S, Shah VB, Edelman A (2012) Julia: A fast dynamic language for technical computing. arXiv:1411.1607.

Bixby RE (2012) A brief history of linear and mixed-integer programming computation. *Documenta Mathematica, Extra Volume: Optim. Stories* 107–121.

Bühlmann P, Van De Geer S (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications* (Springer, Berlin).

Chatterjee S, Hadi AS, Price B (2012) *Regression Analysis by Example*, 5th ed. (John Wiley & Sons, New York).

Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. *SIAM J. Scientific Comput.* 20(1):33–61.

DeCock D (2011) Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *J. Statist. Ed.* 19(3):1–15.

DiCiccio TJ, Efron B (1996) Bootstrap confidence intervals. *Statist. Sci.* 189–212.

Donoho DL, Huber PJ (1983) The notion of breakdown point. *A Festschrift for Erich L. Lehmann* 157–184.

Draper NR, Smith H (1998) *Applied Regression Analysis*, 3rd ed. (John Wiley & Sons, New York).

Efron B (1979) Bootstrap methods: Another look at the jackknife. *Ann. Statist.* 7(1):1–26.

Eldar YC, Kutyniok G (2012) *Compressed Sensing: Theory and Applications* (Cambridge University Press, London).

Furnival GM, Wilson RW (1974) Regressions by leaps and bounds. *Technometrics* 16(4):499–511.

Greenshtein E (2006) Best subset selection, persistence in high-dimensional statistical learning and optimization under l1 constraint. *Ann. Statist.* 34(5):2367–2386.

Gurobi Inc. (2014) Gurobi optimizer reference manual. Accessed August 20, 2014, http://www.gurobi.com.

Hampel FR (1971) A general qualitative definition of robustness. *Ann. Math. Statist.* 42(6):1887–1896.

Hastie T (2015) Trevor Hastie lectures and talks. Accessed February 11, 2015, http://www-stat.stanford.edu/~hastie/TALKS/glmnet_webinar_Rsession.tgz.

Jacob L, Obozinski G, Vert J-P (2009) Group Lasso with overlap and graph lasso. *Proc. 26th Ann. Internat. Conf. Machine Learn.* (ACM, New York), 433–440.

Javanmard A, Montanari A (2013) Confidence intervals and hypothesis testing for high-dimensional regression. arXiv preprint arXiv:1306.3171.

Lockhart R, Taylor J, Tibshirani RJ, Tibshirani R, et al. (2014) A significance test for the Lasso. *Ann. Statist.* 42(2):413–468.

Lubin M, Dunning I (2015) Computing in operations research using Julia. *INFORMS J. Comput.* 27(2):238–248.

Ma S, Song X, Huang J (2007) Supervised group lasso with applications to microarray data analysis. *BMC Bioinformatics* 8(1):60.

Massy WF (1965) Principal components regression in exploratory statistical research. *J. Amer. Statist. Assoc.* 60(309):234–256.

Mazumder R, Friedman JH, Hastie T (2011) Sparsenet: Coordinate descent with nonconvex penalties. *J. Amer. Statist. Assoc.* 106(495):1125–1138.

Miller A (1990) *Subset Selection in Regression* (CRC Press, Boca Raton, FL).

Nemhauser G (2013) Integer programming: The global impact. *EURO, INFORMS, Rome, Italy*, http://euro2013.org/wp-content/uploads/Nemhauser_EuroXXVI.pdf.

R Core Team (2014) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org/.

Raskutti G, Wainwright MJ, Yu B (2011) Minimax rates of estimation for high-dimensional linear regression over-balls. *IEEE Trans. Inform. Theory* 57(10):6976–6994.

Rhee SY, Taylor J, Wadhera G, Ben-Hur A, Brutlag DL, Shafer RW (2006) Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proc. Natl. Acad. Sci.* 103(46):17355–17360.

Rousseeuw PJ (1984) Least median of squares regression. *J. Amer. Statist. Assoc.* 79(388):871–880.

Ryan TP (2008) *Modern Regression Methods*, Vol. 655 (John Wiley & Sons, New York).

Seber GA, Lee AJ (2003) *Linear Regression Analysis*, 2nd ed. (John Wiley & Sons, New York).

Shen X, Pan W, Zhu Y, Zhou H (2013) On constrained and regularized high-dimensional regression. *Ann. Institute Statist. Math.* 65(5):807–832.

Tabachnick BG, Fidell LS (2001) *Using Multivariate Statistics* 4th ed. (Allyn and Bacon, Boston).

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

Top500.org (2013) Top500 Supercomputer Sites, Directory page for Top500 lists. Result for each list since June 1993. Accessed December 4, 2013, http://www.top500.org/statistics/sublist/.

Torgo L (2014) Regression data sets. Accessed August 20, 2014, http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html.

Tsanas A, Xifara A (2012) Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings* 49:560–567.

RIGHTSLINK

Weisberg S (2014) *Applied Linear Regression*, 4th ed. (John Wiley & Sons, New York).

Winner L (2014) Miscellaneous data sets. Accessed August 20, 2014, http://www.stat.ufl.edu/~winner/datasets.html.

Wold S, Ruhe A, Wold H, Dunn W III (1984) The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Scientific Statist. Comput.* 5(3):735–743.

Xu H, Caramanis C, Mannor S (2009) Robustness and regularization of support vector machines. *J. Machine Learn. Res.* 10:1485–1510.

Yeh IC (1998) Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Res.* 28(12): 1797–1808.

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

Zhang C-H, Zhang T (2012) A general theory of concave regularization for high-dimensional sparse estimation problems. *Statist. Sci.* 27(4): 576–593.

Zhang Y, Wainwright MJ, Jordan MI (2014) Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. *J. Machine Learning Research: Workshop and Conf. Proc.* 35:1–28.

Zhao P, Rocha G, Yu B (2009) The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Statist.* 37(6A): 3468–3497.

Dimitris Bertsimas is the Boeing Professor of Operations Research and the co-director of the Operations Research Center at the Massachusetts Institute of Technology. The present paper is part of his 2013 Morse lectureship and part of his research on classical statistics problems under a modern optimization lens.

Angela King received her Ph.D. at the Operations Research Center at the Massachusetts Institute of Technology in 2015 under the supervision of Dimitris Bertsimas. Her doctoral thesis addresses regressions problems under a modern optimization lens.

Downloaded from informs.org by [98.217.202.234] on 13 February 2016, at 09:26 . For personal use only, all rights reserved.

RIGHTSLINK