An Algorithmic Approach to Linear Regression

Bachelor Thesis[1]

Menno van Beek[2]

Supervisor : R. Hoogervorst[3] MSc
Co-reader : dr. K.S. Postek[4]

June 29, 2018

[1] Erasmus University Rotterdam, Erasmus School of Economics, Econometrics and Operations Research.

[2] Student ID: 412261.

[3] PhD Candidate at Erasmus University Rotterdam (Department of Econometrics).

[3] Assistant Professor at Erasmus University Rotterdam (Department of Econometrics).

Abstract

We analyze a procedure how to build a high quality linear regression model. We start with an overview of the desirable properties, for example robustness and limited pairwise multicollinearity, of the linear regression model. We discuss the problem that the current approaches are not capable to find a linear regression model with the desirable properties. Therefore, our goal is to find a procedure that produces a linear regression model which achieves the desirable properties in a reasonable amount of time. We present an algorithmic approach in which the desirable properties are modeled as constraints and through penalties in the objective function of a Mixed Integer Quadratic Optimization (MIQO) model. The performance of the algorithm is shown on both real and synthetic data sets, and is compared with the widely used Lasso approach from Tibshirani (1996). Lastly, we extent the MIQO model with a heuristically chosen subset of interaction terms and compare the performance of the heuristic with Lasso on a synthetic data set.

Keywords: Linear Regression Model, Mixed Integer Quadratic Optimization, Interaction Terms

Acknowledgements

This thesis was written as part of the Bachelor's degree programme Econometrics and Operations Research at the Erasmus University Rotterdam. I would like to express my appreciation to my supervisor R. Hoogervorst MSc for his guidance and valuable suggestions throughout the research and writing process, which have greatly helped me to shape this thesis.

Contents

1 Introduction

The linear regression model considers the relationship between a dependent variable $\boldsymbol{y} \in \mathbb{R}^n$ and a matrix of explanatory variables $\boldsymbol{X} \in \mathbb{M}^{n \times p}$. The associated parameter $\boldsymbol{\beta} \in \mathbb{R}^{p \times 1}$ estimates the effect of $\boldsymbol{X}$ on $\boldsymbol{y}$. Lastly, the error terms $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times 1}$ represent the part of $\boldsymbol{y}$ that is not explained by the explanatory variables. The linear regression model is given by

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

The linear regression model is often used by data scientists. The power of this model lies in its simplicity and interpretability. Thereby, the desirable properties, for example robustness and limited pairwise multicollinearity, of the linear regression model are widely studied in the literature. Common textbooks provide numerous variable selection procedures to find a linear regression model which has the desirable properties built in.

However, these textbooks primarily focus on building in the desirable properties one at time, instead of jointly. This naturally leads to iterative approaches in which the modeler selects explanatory variables according to a specified criterion and performs a series of checks to see if the linear regression model has the desirable properties. Especially when the number of explanatory variables is large, these iterative approaches are not capable to produce a high quality linear regression model in a reasonable amount of time. Finally, the modelers approach is decisive for the resulting linear regression model, and hence there is no guarantee that the resulting linear has the desirable properties of a linear regression model.

1.1 Contribution and Structure of this Thesis

In this thesis we will elaborate on the paper Bertsimas and King (2015), in which the authors proposed an algorithmic approach to find a linear regression model that balances the desirable properties of a linear regression model simultaneously. To ensure that the resulting linear regression model has the desirable properties, we formulate the desirable properties of a linear regression model as constraints and through penalties in the objective function of a Mixed Integer Quadratic Optimization (MIQO) problem. To measure the quality of the resulting model we use out-of-sample R^2 and to what extent the model reaches interpretability, significance, robustness to error in data, and sparsity. Furthermore, we compare the performance of the algorithm with the Lasso approach from Tibshirani (1996), which is a robust variable shrinkage method to find a linear regression model. In the last part of this thesis we extent the algorithmic approach by including a heuristically chosen subset of interaction terms.

This thesis is structured as follows. In §2 we describe the desirable properties of a linear regression model in detail and give the problem description. The corresponding literature is reviewed in §3. We outline the MIQO based algorithm in detail in §4. In §5 we show computational results of the algorithm and compare the algorithm with Lasso. The methodology and results of the interaction terms are described in §6. We conclude in §7.

2 Problem Description

In this section, we review the desirable properties of a linear regression model which are: general sparsity, group sparsity, limited pairwise multicollinearity, nonlinear transformations, robustness, stability to outliers, modeler expertise, statistical significance and low global multicollinearity. Thereby, we consider how each property is currently taken into account by a modeler and how we can take each property into account in our approach. Lastly, we define the problem of finding a high quality linear regression model, which has the desirable properties balanced, in a reasonable amount of time.

2.1 Desirable Properties of a Linear Regression Model

General Sparsity

When the number of explanatory variables is high, a modeler would like to construct a linear regression model in which only the most relevant explanatory variables for the response are included. In the literature, this is known as identifying a critical subset of the response (see Miller (2002)). A linear regression model with only the most relevant explanatory variables is more interpretable and the prediction error decreases because of the elimination of noise variables. Therefore, we want to construct linear regression models with a specific number k of nonzero regression coefficient $\boldsymbol{\beta}$. We call the number k the general sparsity of the model. To model this in our approach, we restrict the maximum number of nonzero regression coefficients $\boldsymbol{\beta}$ by k and solve our model for all possible values of k.

Group Sparsity

In some linear regression models there are explanatory variables that have a group structure, which is a set of explanatory variables that are coherent to each other. Categorical variables that are expanded to a set of dummy variables naturally form a group structure. To make sure that the variables in a group structure interpretable, a modeler naturally includes either all or none of the variables from the group structure. A common approach to preserve a group structure is group Lasso (see Yuan and Lin (2006)). Therefore, in our approach we focus on preserving the group sparsity property by restricting that the explanatory variables in a group structure have either all zero coefficients or none.

Limited Pairwise Multicollinearity

Instable parameter estimates can be caused by including a pair of variables that is highly correlated (see Tabachnick and Fidell (2001)). This means that if a pair of variables is highly correlated, the resulting linear regression model could indicate an inaccurately estimated effect of an explanatory variable with the the dependent variable. Normally a modeler would check the correlation matrix to see if the linear regression model has pairs of highly correlated variables. From a modelers perspective checking all pairwise correlation is practically impossible when the number of explanatory variables is high. To avoid instable parameter estimates in our approach, we restrict the maximum pairwise correlation of each pair of variables with nonzero regression coefficients $\boldsymbol{\beta}$.

Nonlinear Transformations

In some applications the dependent variable has a nonlinear relationship with an independent variable. Commonly used nonlinear transformations of an explanatory variable x are $\log(x)$, x^2, $\sqrt{x}$. Normally a modeler would search for such relationships by checking pairwise plots of the dependent and independent variables. When the number of independent variables is high, checking all pairwise plots is practically impossible for a modeler. To model the nonlinear relationship in our approach, we add nonlinear transformations of the explanatory variables. To ensure that the resulting model is sparse, we require that the final model has at most one of the nonlinear transformations or the original explanatory variable itself with a nonzero regression coefficient $\boldsymbol{\beta}$ incorporated.

Robustness

Collected data is often inaccurate. For example, the explanatory variable could be measured with an error. A commonly used approach to take this problem into account is robust optimization. That is, a modeler formulates uncertainty sets to make the resulting model robust to worst-case uncertainty (see Ben-Tal et al. (2009) and Bertsimas et al. (2011)). To make our model immune to worst-case uncertainty, we regularize the objective of our model as in Bertsimas and King (2015).

Stability to Outliers

The collected data may contain outliers which could lead to an incorrect generalization of the model. The linear regression model penalizes the errors quadratically and, therefore, this model is known for its sensitivity against outliers. In the literature one of the proposed solutions is to change the quadratic objective to the least median of squares (LMS) objective. This objective is introduced in Rousseeuw (1984) and it minimizes the median of the l_2-norm of the residuals. The result is that outliers have less effect on the estimated parameter. In our approach, the LMS objective can be used instead of the regular least squares objective if the modeler suspects outliers in the data.

Modeler Expertise

When a modeler has expertise in the field of research where the data comes from, he might want to specify explanatory variables that have to be included in the linear regression model. The reason that a modeler includes a set of explanatory variables is based on the intuition from the modeler that this set has a known correlation with the dependent variable. We take modeler expertise into account in our approach by ensuring that the specified set of explanatory variables has to be included in the final linear regression model.

Statistical Significance

Good linear regression models truly detect the relationship between the explanatory variables and the response. A commonly used approach in the literature is to use the concept of statistical significance. An explanatory variable x is considered to be statistically significant if the probability that the observed effect happens by chance is less than the significance level α, in presence of the other explanatory variables (a typical value for α is 5%). A modeler naturally removes insignificant explanatory variables from the final linear regression model because the interpretation of the effect of insignificant variables is obscure. In our approach we take statistical significance into account by ensuring that our final only contains significant variables. However, we would like our approach to be free of any distributional assumption, so that high dimensional settings can be handled and the regularization (see Robustness) can be incorporated. Therefore, we use residual bootstrapping (see Efron (1982)) to estimate the significance of the regression coefficients, which is asymptotically more accurate than using the standard normality assumptions (see DiCiccio and Efron (1996)).

Low Global Multicollinearity

Besides the pairwise multicollinearity problem, modelers also face the problem of global multicollinearity (see Ryan (2008) for an example). Global multicollinearity refers to the situation in which one explanatory variable can be linearly predicted by two or more other explanatory variables. The result of global multicollinearity is that it leads to instable parameter estimates, which means that a small change in the data or linear regression model can heavily change the parameter estimates. In other words, the restriction on pairwise multicollinearity may not be sufficient to cover the problem of global multicollinearity. Therefore, a modeler normally performs a check on this property. A commonly used check is to calculate the condition number of the matrix of explanatory variables that appear with a nonzero regression coefficient $\boldsymbol{\beta}$ in the final model. In Chatterjee and Hadi (2015) the authors argue that a condition number greater than 30 is taken as evidence of multicollinearity. Hence, when there is evidence of multicollinearity a modeler has to resolve this problem by excluding variables from the model to make sure that the parameter estimates are stable.

2.2 Finding a Linear Regression Model with the Desirable Properties

When building a regression model the modeler has to balance the desirable properties of a high quality regression model, which are typically built in the model one at a time. Common regression textbooks outline variable selection approaches together with a series of checks how a good regression model can be found, but provide almost no guidance which approach the modeler should use in a particular setting. Hence, it can be expected that two modelers with the same data set give different models only because they used a different approach. This is mainly because the proposed approaches in most textbooks try to find the model iteratively in which the properties are built in one at time. The result is that one approach may find that the desired properties cannot be built in the model all together while another approach could result in a model that does have the desired properties. In other words, the iterative procedure of refinements applied by the modeler is decisive for the resulting model.

From a modelers perspective the only practically useful approach to built the desirable properties of a linear regression model into his model is an iterative procedure of refinements. The reason for this is, that there is no practically workable approach that results directly in a model that balances the competing properties at once in a reasonable amount of time.

Although the current approaches from regression textbooks are able to address each property one at a time, the problem with these approaches remains that this could lead to models that do not satisfy the desirable properties because the properties are balanced iteratively instead of simultaneously. Furthermore, when the potential number of explanatory variables is high the current approaches fail because they are practically unworkable. Hence, we focus in this thesis on the problem of finding a high quality linear regression model that balances the desirable properties of the linear regression model jointly in a reasonable amount of time.

3 Literature Review

The linear regression model is extensively studied in the literature. Commonly used linear regression textbooks (see for example Heij et al. (2004), Chatterjee and Hadi (2015)) pay attention to the assumptions and limitations of this model. These textbooks provide approaches with variable selection methods, for example the top-down approach from Heij et al. (2004) in which the modeler starts with a linear regression model with all explanatory variables included and eliminates variables based on some specific criteria (for example statistical significance). However, these textbooks provide more approaches than the top-down approach without offering guidance which approach should be used. Furthermore, most approach are practically useless if the number of explanatory variables is high. Therefore, building a linear regression model that addresses the desirable properties in a reasonable amount of time is more of an art than science. For example, to achieve general sparsity with a maximum of k nonzero regression coefficients (of a total of p possible explanatory variables) a possible approach is to solve for all values of k the following problem:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left|\left| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \right|\right|_2^2 \right\}, \qquad \text{subject to: } \left|\left| \boldsymbol{\beta} \right|\right|_0 \leq k. \tag{1}$$

This is more generally known as the best subset selection problem (see Miller (2002) for details). An efficient way to solve (1) is proposed by Furnival and Wilson (1974), which is MIQO based. However, the cardinality constraint on $\boldsymbol{\beta}$ makes the problem NP-hard (Natarajan (1995)). Therefore, state-of-the-art algorithms, for example `leaps` in `R` (RStudio Team (2015)), can only solve the problem accurately for $p \leq 30$. Hence, the approach to solve the best subset problem to find a good linear regression model is considered to be inaccurate for $p > 30$. Therefore, a common approach considered in the literature is a surrogate of (1) which is known as Lasso (Tibshirani (1996), Chen et al. (2001)). Although Lasso can be used in more general models than the linear regression model, we consider the linear regression form of Lasso, which is to solve

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

In Bertsimas et al. (2011) the authors compare Lasso with the best subset problem and conclude that Lasso does not produce sparse models in general, but that the predictive power is comparable. The reason that Lasso predicts well is because it is a robust model against uncertainty in data (Bertsimas and King (2015)). Although the predictive power of Lasso is reasonably good, the desirable properties are not guaranteed and mostly not achieved because the regularity conditions (Bühlmann and Van De Geer (2011)) are violated. Therefore, we focus in this thesis on an approach to find a high quality linear regression model that achieves the desirable properties in a reasonable amount of time and compare our approach with Lasso in terms of predictive power and interpretability.

4 MIQO Based Approach

In this section, we describe the algorithm from Bertsimas and King (2015) to solve the problem formulated in §2.2, which is to find a high quality linear regression model that balances the desirable properties of the linear regression model jointly in a reasonable amount of time. This problem naturally lends itself to be formulated as a Mixed Integer Quadratic Optimization (MIQO) model. The desirable properties of a linear regression model as described in §2 can be formulated as integer constraints or can be taken into account in the objective function through penalties. The algorithm from Bertsimas and King (2015) is decomposed into three stages. In the first stage of the algorithm, we pre-process the data and compute all the parameters necessary for our MIQO approach. In the second stage we solve the MIQO model. After we have solved the MIQO model, we go to the last step of our algorithm and generate additional constraints if the solution of our MIQO does not satisfy the desirable properties and go back to step 2.

4.1 Stage 1: Preprocessing and Computing Parameters

In the first stage the data is split randomly (50% / 25% / 25%) into training, validation and test set. We standardize each data set column wise based on the mean and variance of columns of the training set, so that each column has zero mean and unit l_2 norm. When the data is standardized, the modeler may specify the number of robustification parameters Γ to be tested (the default is 5) and the maximum pairwise correlation ρ that is allowed between the explanatory variables (the default is 0.80).

The algorithm then proceeds to generate the correlation matrix based on the training set and generates $\mathcal{HC}$, the set of pairs of variables that have a pairwise correlation in absolute value beyond ρ. At this point, the algorithm requires the modeler to specify which explanatory variables have a natural group structure. The algorithm then expands categorical variables to dummy variables and makes sets of group variables. The mth set explanatory variables that require a group structure is denoted by $\mathcal{GS}_m$. Furthermore, the modeler also specifies which nonlinear transformations have to be considered, where we denote the mth set of nonlinear transformations by $\mathcal{T}_m$ (which also includes the non-transformed variable). If the modeler has some subjective expertise about the variables, the modeler can specify this at this point in a set of explanatory variables that have to be present in the final linear regression model, where this set is denoted as $\mathcal{J}$. Lastly, if the modeler suspects outliers he can specify to use the median (LMS) objective instead of the regular least squares objective.

The algorithm proceeds to calculate $k_{\max}$, the maximum number of variables that can be included in the final model without violating the constraint on maximum pairwise correlation. We determine $k_{\max}$ by solving an independent set problem, in which the explanatory variables are represented by nodes and the edges by pairs of variables. The binary indicator variable in this independent set problem is z_i, where $z_i = 1$ indicates that explanatory variable x_i is included in the independent set problem and z_i is zero otherwise. The constraints on maximum pairwise correlation are incorporated by excluding edges for which the pairs of variables have a pairwise correlation beyond ρ. The independent set problem is given by

$$
\begin{aligned}
k_{\max} = \max_{\boldsymbol{z}} & \left\{ \sum_{i=1}^{p} z_i \right\}, && && (3) \\
\text{s.t.} \qquad & z_i + z_j \leq 1, && \forall\, (i,j) \in \mathcal{HC}, && (4) \\
& z_i \in \{0,1\}, && \forall\, i = 1, \ldots, p. && (5)
\end{aligned}
$$

Thereafter, the algorithm determines the values Γ to be tested as follows: We denote $\mathcal{RB}$ as the set of logarithmically spaced values of Γ between 0 and the value of Γ for which the optimal solution of the unconstrained problem is $\boldsymbol{\beta} = \mathbf{0}$. The value for which $\boldsymbol{\beta} = \mathbf{0}$ can be found by applying a coordinate descent algorithm. The algorithm has now computed all relevant parameters and proceeds to Stage 2.

4.2 Stage 2: The MIQO Model

In the second stage of the algorithm we solve the MIQO model for all combinations (k, Γ), where $k \in \{1, \ldots, k_{\max}\}$ and $\Gamma \in \mathcal{RB}$, using the training data $(\boldsymbol{y}, \boldsymbol{X})$. The MIQO model is as follows:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}, \boldsymbol{z}} \quad & \left\{ \frac{1}{2} \left|\left| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \right|\right|_2^2 + \Gamma \left|\left| \boldsymbol{\beta} \right|\right|_1 \right\}, && && (6) \\
\text{s.t.} \quad & -\mathcal{M} z_l \leq \boldsymbol{\beta}_l \leq \mathcal{M} z_l, && \forall\, l = 1, \ldots, p, && (7) \\
& \sum_{l=1}^{p} z_l \leq k, && && (8) \\
& z_i + z_j \leq 1, && \forall\, (i,j) \in \mathcal{HC}, && (9) \\
& z_1 = \ldots = z_l, && (1, \ldots, l) \in \mathcal{GS}_m,\ \forall\, m, && (10) \\
& \sum_{i \in \mathcal{T}_m} z_i \leq 1, && \forall\, m, && (11) \\
& z_l = 1, && \forall\, l \in \mathcal{J}, && (12) \\
& \sum_{l \in \mathcal{S}_i} z_l \leq |\mathcal{S}_i| - 1, && \forall\, \mathcal{S}_1, \ldots, \mathcal{S}_j && (13) \\
& z_l \in \{0,1\}, && \forall\, l = 1, \ldots, p, && (14)
\end{aligned}
$$

In the objective (6), the first term minimizes the residual sum of squares and the second term is the robustification penalty on the parameter $\boldsymbol{\beta}$. The robustification parameter makes the resulting model robust against uncertainty in the data. In constraints (14) a binary variable z_l is introduced. This binary indicator is used in constraints (7) to ensure that the parameter $\boldsymbol{\beta}_l$ is only non-zero if $z_l = 1$. Furthermore, in constraints (7) the constant $\mathcal{M}$ denotes the maximum value the parameter $\boldsymbol{\beta}_l$ can take in absolute value.

The constraints (8)−(13) are used to guarantee the desirable properties of the linear regression model. To ensure general sparsity, we restrict the number of nonzero regression coefficients in by k in constraint (8). To avoid the issue of high pairwise multicollinearity we restrict the pairwise correlation of each pair of explanatory variables in the final model by including the set of constraints (9). To ensure that the parameters for a group structure are either all zero or not, we require in (10) that all indicator variables z_l in the mth group structure $\forall\, l \in \mathcal{GS}_m$ are equal. Constraints (11) ensure that for each set of transformed explanatory variables there may only appear one of the transformed variables or the original variables itself in the final linear regression model. If the modeler specified a set of variables that should always appear in the resulting model, we generate the constraints (12) which ensures that the indicator variables z_l with indices $l \in \mathcal{J}$ are equal to 1. The last set of constraints (13) is initially not present in the model, but might be generated in Stage 3 if the regression model corresponding with solution $\mathcal{S}_i$ has condition number greater than 30 or if the parameters are insignificant. Note that both the condition number and significance can not be incorporated directly as constraints, because both the calculation of the condition number as well as bootstrapping the parameters involve highly nonlinear calculations.

For each combination of k and Γ the output of the MIQO model is a set of regression parameters $\boldsymbol{\beta}^*$ and a set of indicator variables $\boldsymbol{z}^*$. The algorithm then calculates for each combination of k and Γ the out-of-sample R^2 based on the validation set, which is based on the following definition

Definition 1 *Let $\boldsymbol{y}_{in}$ denote the vector of in-sample response data and let $\boldsymbol{y}_{out}$ denote vector of out-of-sample response data. Furthermore, let $\hat{y}_i$ denote the prediction of $y_i \in \boldsymbol{y}_{out}$ and let $\bar{y}$ be the mean $\boldsymbol{y}_{in}$. Then the out-of-sample R^2 is defined by*

$$R^2 = 1 - \frac{\sum_{y_i \in \boldsymbol{y}_{out}} (y_i - \hat{y}_i)^2}{\sum_{y_i \in \boldsymbol{y}_{out}} (y_i - \bar{y})^2} \tag{15}$$

The importance of measuring predictive power based on out-of-sample R^2 is shown in Campbell and Thompson (2007), in which the authors predict excess stock returns. As the authors predict time series, we alter their definition of out-of-sample R^2 so that we can use it on regular linear regression data.

Finally, when all combinations of k and Γ are considered, the algorithm chooses the three best regression models (denoted by $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$) based on the three highest out-of-sample R^2 values and proceeds to Stage 3, where additional constraints might be generated.

4.3 Stage 3: Generating Additional Constraints

For each of the solutions $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$, the final step is to check if all the regression coefficients are statistically significant based on bootstrapping and if there is evidence of global multicollinearity. We adopt the residual bootstrap method from Efron (1982), in which the residuals from the final regression model are used to bootstrap the regression coefficients. For each regression coefficient we use as significance level $\alpha = 5\%$. Evidence of multicollinearity is indicated by a condition number greater than 30. Each solution $\mathcal{S}_i$ for which these two properties are not satisfactorily addressed is excluded from the set of possible linear regression models by including constraint (13). Constraint (13) cuts solution $\mathcal{S}_i$ of the binary hypercube by requiring that the corresponding solution to $\mathcal{S}_i$ of indicator variables $\boldsymbol{z}$ is infeasible in next iterations. The algorithm returns back to Stage 2 as long as the algorithm has not found a top three of best linear regression models, and the number of iterations allowed between Stage 2 and 3 (default is 3) is not reached. The algorithm terminates when it finds a top three of solutions in which the last two properties are satisfactorily addressed or when the number of iterations allowed between Stage 2 and 3 is reached. The output of this algorithm is thus a top three of linear regression models, in which the desirable properties of the linear regression model are jointly balanced.

5 Computational Experiments

In this section, we illustrate the performance of our algorithm on both synthetic and real data sets. We consider synthetic data sets with no special structure to demonstrate that our algorithm achieves interpretability and robustness. Thereby, we also show the performance of our algorithm on basic real data sets with no special structure. Furthermore, we consider a real data set in which it is known that the explanatory variables have a nonlinear relationship with the response. We also consider a real data set in which there are explanatory variables that are categorical to show the performance of the algorithm on the group sparsity property. Lastly, we compare our algorithm with Lasso to show that our algorithm achieves, in general, more interpretable models than Lasso and results in a linear regression model that has the desirable properties with predictive power comparable Lasso.

Synthetic Data

We generated synthetic data where each observation $i = 1, \ldots, 2n$ is an independent realization from an p-dimensional multivariate normal distribution with zero mean and covariance matrix $\Sigma := \sigma_{ij}$. That is, we generated $\boldsymbol{x}_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \Sigma)$. We took $\sigma_{ij} = \rho^{|i-j|}$ and varied ρ in the experiments as $\rho \in \{0, 0.8, 0.9\}$. For each experiment we generated the error term as $\epsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2)$, for a σ^2 that varies as $\sigma^2 \in \{0.5,\ 1,\ 2\}$. The vector of response $\boldsymbol{y}$ is generated as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where we equally spaced the k (we consider $k = 10$) nonzero β_i's and thus take $\beta_i = 1$ if i mod $p/k = 0$ and zero otherwise. Lastly, to show the algorithms performance on robustness we add a measurement error $\boldsymbol{\Delta X}$ to $\boldsymbol{X}$. In each experiment we report how the measurement error is generated.

Real Data

We tested the algorithm on eleven publicly available data sets. We obtained the data sets CPU, Yacht Hydrodynamics, White Wine Quality and Red Wine Quality from the UCI Machine Learning repository (Dheeru and Karra Taniskidou (2017)). The wine quality data sets originally come from Cortez et al. (2009). The data sets Compact, Elevator and Pyrimidines were obtained from the University of Porto (Torgo (2014)). The data sets LPGA 2008, LPGA 2009 and Airline Costs were obtained from the University of Florida (Winner (2014)). The Diabetes data is obtained from the `Lars` package in `R` (RStudio Team (2015)).

Computational Specifications

All computations were performed on a MacBook Air computer with an Intel Core i7 3667U (2.0 GHz) processor and 8 GB of RAM. We used `Gurobi 8.0.0` (Gurobi Optimization (2016)) to optimize the MIQO problems and implemented the algorithm in `Java`. We used `MATLAB` (MATLAB (2017)) to compute Lasso solutions. To compute group Lasso solutions we used the `grplasso` package in `R` (RStudio Team (2015)).

5.1 Basic Structure Data

In this section we show the performance of the algorithm on real and synthetic data sets. Although the algorithm generates the top three solutions, we only report the linear regression model with the highest out-of-sample R^2 on the validation set. Note that for each data set our algorithm requires the modeler to specify the number of iterations allowed between Stage 2 and 3. The result is that the number of MIQO problems to be solved is $k_{\max} \times \#$ of regularization parameters to be tested $\times$ # of iterations allowed between Stage 2 and 3 MIQO problems. However, in our experiments on the synthetic data sets we altered the algorithm because the computation time was unreasonably high if we would allow iterations between Stage 2 and 3. In the experiments on real data sets we use the algorithm as described in §4.

We altered the algorithm to merge Stage 2 and 3 as follows: For each MIQO problem with input variables (k, Γ) and output variables ($\boldsymbol{\beta}^*$, $\boldsymbol{z}^*$) we calculate the out-of-sample R^2 based on the validation set. If this R^2 is higher than at least one of the top three solutions $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that the algorithm has found so far, we calculate the condition number and bootstrap the regression coefficients. When there is no evidence of multicollinearity and all regression coefficients are statically significant, we update the solution $\mathcal{S}_i$ with the lowest value of R^2 based on the validation set with the found solution.

This results in less computation time because the calculation of the condition number and bootstrapping method are relatively fast. Also note that the calculation of both the condition number and bootstrapping is only necessary for improvements of the R^2 value in comparison with the best solution obtained at that specific moment. The altered algorithm produces linear regression models that achieve the desirable properties in a reasonable amount of time and which are still selected based on the highest values of out-of-sample R^2 on the validation set. Although, the altered algorithm results in less computation time we also see the disadvantage of merging Stage 2 and 3. In the worst-case scenario this could lead to models with less predictive power. More specifically, for some k we could obtain in the worst-case scenario that for all robustification parameters the underlying pattern of k nonzero regression coefficients is not identified because the corresponding model has a too high condition number of statistically insignificant regression coefficients, which would only have been possible if the corresponding solutions were excluded.

In Tables 1 - 6 we present the results on the synthetic data sets. Each row in each table corresponds with one experiment on one synthetic data set. For each experiment we report: the variance of the error terms (σ^2), the robustification parameter selected by our algorithm (MIQO Γ^*), number of nonzero regression coefficients selected by the algorithm (MIQO K^*), the number of true nonzero regression coefficients identified by the algorithm (TP), the out-of-sample R^2 based on the test set, the maximum pairwise correlation of two explanatory variables with nonzero regression coefficients (MaxCor), the condition number (Cond). We also report for each experiment the corresponding results for the Lasso approach except for the robustication parameter, which is selected according to the highest value of the out-of-sample R^2 based on the validation set. Lastly, we present the computation time in hours for each experiment, which serve to show that the algorithm can be used in practice in a reasonable amount of time.

In Tables 1 and 2 we present the performance of the algorithm on the general sparsity property for both the regular case $n < p$ as well as the over-identified case $n > p$. Both the algorithm as the Lasso approach identify all true nonzero regression coefficients. The algorithm has comparable predictive power with Lasso based on the out-of-sample R^2 on the test set. However, Lasso brings in additional noise variables whereas our algorithm achieves to identify the exact pattern.

Tables 3 and 4 illustrate the performance of the algorithm on the limited pairwise multicollinearity property, for $n < p$ and $n > p$. Again, both the algorithm as well as the Lasso approach identify all true nonzero regression coefficients, with a comparable predictive power, and where Lasso brings in additional noise variables. Note that also the algorithm brings in two noise variables for the specific case $n < p$ and $\sigma^2 = 2$, which might be the result of random effects. However, the advantage of the algorithm over the Lasso approach becomes more clear when we compare the maximum pairwise correlation and condition number of both approaches. Here, we see that the Lasso approach produces linear regression models with both a high pairwise correlation and a high condition number, which is the result of bringing in additional noise variables. In contrast, the algorithm produces linear regression models without violating these desirable properties, so that we we avoid instable parameter estimates. Lastly, we note that in Table 3 the computation time is relatively low compared to the other tables, which is the result of less MIQO problems to be solved. Review that the number of MIQO problems is a linear function of $k_{\max}$ for a fixed number of robustification parameters to be tested. The maximum pairwise correlation allowed is by default 0.80, which means that the algorithm finds a much lower value of $k_{\max}$ in Stage 1 of the algorithm compared to the other tables.

Tables 5 and 6 are designed to show the performance of the algorithm on the robustness property. Again we consider both $n < p$ and $n > p$. The algorithm and Lasso perform equal in identifying the true nonzero parameters and we see again that Lasso brings in additional noise variables in contrast with the algorithm. We note that the algorithm achieves a higher R^2 in all the six experiments in Tables 5 and 6. However, we cannot conclude from this that the algorithm performs better in terms of predictive power, because we see that the R^2 values differ not significantly and we are aware of the fact that we only used one experiment per R^2 value.

In general we clearly see the advantage of the algorithm over MIQO in Tables 1 - 6 because the algorithm performs better to achieve general sparsity, limited pairwise multicollinearity, statistically significant regression coefficients and low global multicollinearity. Lastly, note that the experiments where $n < p$ have relatively high computation times compared to the experiments where $n > p$, which is due to the increase in dimensionality of the MIQO problem but also because in Stage 1 the algorithm finds higher values of $k_{\max}$, and hence more MIQO problems have to be solved.

Table 1: Sparsity

$n = 500$, $p = 100$, $\rho = 0$, $\boldsymbol{\Delta X} = \mathbf{0}$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.091	10	10	0.947	0.138	1.564	4.535	27	10	0.945	0.157	2.432
1	0.004	10	10	0.922	0.127	1.570	4.601	37	10	0.918	0.153	2.811
2	0.004	10	10	0.847	0.107	1.643	4.483	46	10	0.848	0.134	3.092

Table 2: Sparsity

$n = 100$, $p = 500$, $\rho = 0$, $\boldsymbol{\Delta X} = \mathbf{0}$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.094	10	10	0.958	0.231	3.031	5.475	61	10	0.918	0.378	44.167
1	0.021	10	10	0.904	0.237	2.349	5.543	75	10	0.811	0.306	152.400
2	0.077	10	10	0.882	0.285	2.945	5.482	52	10	0.729	0.328	32.253

Table 3: Pairwise multicollinearity

$n = 500$, $p = 100$, $\rho = 0.9$, $\boldsymbol{\Delta X} = \mathbf{0}$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.106	10	10	0.976	0.380	4.020	1.664	30	10	0.976	0.907	113.139
1	0.023	10	10	0.952	0.403	4.710	1.877	35	10	0.951	0.910	132.533
2	0.107	10	10	0.891	0.465	4.716	2.064	28	10	0.885	0.916	128.336

Table 4: Pairwise multicollinearity

$n = 100$, $p = 500$, $\rho = 0.8$, $\boldsymbol{\Delta X} = \mathbf{0}$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.095	10	10	0.931	0.213	2.840	7.412	40	10	0.870	0.829	66.679
1	0.091	10	10	0.910	0.238	3.129	7.432	32	10	0.895	0.815	36.677
2	0.020	12	10	0.701	0.729	8.255	7.749	45	10	0.633	0.862	83.967

Table 5: Robustness

$n = 500$, $p = 100$, $\rho = 0$, $\boldsymbol{\Delta X} \sim \text{Uniform}(0, 2)$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.019	10	10	0.740	0.142	2.541	3.543	31	10	0.713	0.146	2.464
1	0.018	10	10	0.598	0.097	1.604	3.551	40	10	0.577	0.149	3.071
2	0.005	10	10	0.634	0.131	1.724	3.410	52	10	0.613	0.160	3.490

Table 6: Robustness

$n = 100$, $p = 500$, $\rho = 0$, $\boldsymbol{\Delta X} \sim \text{Uniform}(0, 1)$.

σ^2	MIQO Γ^*	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.095	10	10	0.857	0.237	3.290	6.728	49	10	0.771	0.302	25.153
1	0.021	10	10	0.905	0.238	2.736	6.419	82	10	0.802	0.317	213.611
2	0.020	10	10	0.796	0.217	2.722	6.544	78	10	0.604	0.340	198.538

In Table 7 we illustrate the performance of the algorithm on real data sets. We report the size of the training data (n), the number of independent variables in the data set (p), the number of explanatory variables in the resulting regression model (K^*) and the maximum pairwise correlation (MaxCor) present in the resulting linear regression model. From this table we see that the algorithm produces linear regression models with less explanatory variables than Lasso, resulting in more interpretable models. Thereby, the algorithm produces linear regression models with a lower maximum pairwise correlation than Lasso, which means that the algorithm avoids the issue of instable parameters estimates. We note that the algorithm performs similar in terms of predictive power to Lasso, which is in line with the results in Tables 1 - 6 on synthetic experiments. Lastly, we note that the reported values of MIQO K^* is lower in each data set than the reported values in Bertsimas and King (2015). A possible explanation for this difference might be the random effects, as both we as well as Bertsimas and King (2015) only use one experiment. Also, we used different programmes to implement the algorithm which might give a different result. However, the overall contrast that the algorithm produces more interpretable models and achieves the desirable properties in comparison with the Lasso approach, is in both our results as well as in Bertsimas and King (2015) clearly visible.

Table 7: Results for basic structure real data sets

Data set	n	p	MIQO K^*	R^2	MaxCor	Lasso K^*	R^2	MaxCor
CPU	105	6	3	0.788	0.682	6	0.847	0.693
Yacht	154	6	1	0.654	NA*	1	0.637	NA*
White	2,449	11	3	0.278	0.450	11	0.290	0.843
Red	800	11	5	0.345	0.432	10	0.347	0.673
Compact	4,096	21	5	0.703	0.574	18	0.729	0.939
Elevator	8,300	18	6	0.812	0.680	15	0.815	0.999
Pyrimidines	37	26	7	0.657	0.782	18	0.635	0.886
LPGA 2008	77	6	2	0.842	0.025	4	0.852	0.224
LPGA 2009	73	11	2	0.911	0.769	10	0.910	0.941
Airline Costs	15	9	1	0.646	NA*	9	0.688	0.972
Diabetes	221	64	6	0.436	0.450	16	0.495	0.612

*Note that in the Yacht data set both the algorithm as the Lasso approach selected only one covariate so that the maximum pairwise correlation is undefined. The same applies for the Airline Costs data set in which the algorithm chose only one covariate.

5.2 Special Structure Data

In this section we compare the algorithm with Lasso on a real data set where nonlinear transformations are included and also on a real data set with a natural group structure.

Nonlinear Transformations

The real data set we consider for nonlinear transformations is the Concrete Strength data set which we obtained from the UCI Learning Repository (Dheeru and Karra Taniskidou (2017)). The dependent variable is the comprehensive strength of concrete and the independent variables are the age of the concrete in days and the ingredients of concrete which are: Cement, Blast furnace slag, Fly ash, Water, Superplasticizer, Coarse aggregate and Fine aggregate.

First, we tested the performance of the algorithm and Lasso without considering nonlinear transformations. The first row of Table 8 gives the results of this experiment. The algorithm selected the covariates Cement, Blast furnace slag, Fly ash and Age. In contrast, Lasso selected all covariates but achieves similar predictive power in terms of out-of-sample R^2, resulting in a less interpretable model. Thereafter, we considered the extended data set in which we included for each independent variable x the nonlinear transformations x^2, $\log(x)$ and $\sqrt{x}$, where $\log(x)$ is changed to $\log(x + 0.0001)$ for the independent variables that can have the value 0. The results of the algorithm and Lasso are given by the second row of Table 8. Our algorithm selected the three linear covariates: Blast furnace slag, Fly ash and Coarse aggregate, and the four nonlinear covariates: $\sqrt{\text{Cement}}$, log(Superplasticizer), log(Fine aggregate) and log(Age). Lasso selected the three linear covariates: Cement, Blast furnace slag and Water, but brings in ten nonlinear covariates (Cement^2, Superplasticizer^2, Fine aggregate^2, Age^2, $\sqrt{\text{Cement}}$, $\sqrt{\text{Blast furnace slag}}$, $\sqrt{\text{Water}}$, log(Fly ash), log(Superplasticizer) and log(Age)), which results in a less interpretable model and has the issue of high pairwise multicollinearity. From this we see that although Lasso selected more explanatory variables, the predictive power is comparable while the algorithm results in a more interpretable model and avoids instable parameter estimates due to pairwise multicollinearity.

Table 8: Nonlinear transformations

Transformation	n	p	MIQO K^*	R^2	MaxCor	Lasso K^*	R^2	MaxCor
Linear	515	8	5	0.632	0.397	8	0.625	0.338
Nonlinear	515	32	7	0.796	0.597	13	0.816	0.999

Group Sparsity

The real data set we consider is the Energy Efficiency data set obtained from the UCI Learning Repository (Dheeru and Karra Taniskidou (2017)), in which we have the two dependent variables that describe building properties: heating load and cooling load. Both dependent variables are considered to be a function of the continuous covariates: Relative compactness, Surface area, Wall area, Roof area, Overall height and Glazing area. Furthermore, the data set also comes with two categorical variables: Orientation (4 categories) and Glazing area distribution (6 categories).

After the algorithm expanded the categorical variables to dummy variables, we compared the performance of the algorithm with group Lasso (Yuan and Lin (2006)). In Table 9 we present results for both dependent variables Heating and Cooling Load. Our algorithm chose to exclude the explanatory variables that have a group structure from the final linear regression model. Group Lasso selected for both dependent variables all covariates, including the group structure, except wall area. Although it is notable that our algorithm excluded the explanatory variables with a group structure, the predictive power is still comparable to group Lasso but results in a more sparse model with a lower pairwise correlation.

Table 9: Group structure

Dependent Variable	n	p	MIQO K^*	R^2	MaxCor	Lasso K^*	R^2	MaxCor
Heating Load	384	14	3	0.901	0.304	13	0.914	0.875
Cooling Load	384	14	3	0.858	0.276	13	0.868	0.880

6 Incorporation of Cross Terms

In some applications the effect of the explanatory variables on the response is based on two or more explanatory variables that interact. See for example Friedrich (1982), in which the authors argue why interaction terms should be considered. In this section we describe the problems that arise when we consider all cross terms of two explanatory variables. Although, cross terms could also be formed with three or more explanatory variables (multi-cross terms), we only consider cross terms of two explanatory variables because of the difficult interpretation that is associated with multi-cross terms. We propose a heuristic to make a subset of cross terms and illustrate the performance of this heuristic based on a synthetic data set.

6.1 Problems with Cross Terms

The results on synthetic data sets in §5.1 show that increasing the number of explanatory variables increases computational time drastically. For a dataset with originally p explanatory variables, including all cross terms would increase the number explanatory variables from p to $\binom{p}{1} + \binom{p}{2} = \frac{1}{2}p^2 + \frac{1}{2}p$. Hence, including all cross terms is from a practical point of view impossible, so that we have to consider a subset of all cross terms. Also, including a cross term based on two explanatory variables x_i and x_j is difficult to interpret if at least one of these two explanatory variables has a regression coefficient that is statistically insignificant when including cross term $x_i x_j$.

6.2 Heuristic Selection of Cross Terms

In §6.1 we argued that cross term $x_i x_j$ should only be included if both explanatory variables x_i and x_j have a nonzero regression coefficient. Therefore, in our heuristic we take advantage of this argument. We propose the following heuristic:

1. In the first step the algorithm from §4 is used to find a linear regression model with the desirable properties for a given data set $(\boldsymbol{y}, \boldsymbol{X})$ where cross terms are not included in $\boldsymbol{X}$. The output of the algorithm consists of the top three best linear regression models $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$, with corresponding vectors of regression coefficients $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2, \boldsymbol{\beta}^3$. Let $\mathcal{I}$ be the set of indices for which at least one of $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2, \boldsymbol{\beta}^3$ has a nonzero regression coefficient. More specific, $i \in \mathcal{I} \Leftrightarrow \exists \beta_i^k \neq 0$ for some $k = 1,\, 2,\, 3$.

2. In the second step we include all cross terms $x_i x_j$ of explanatory variables x_i and x_j $\forall i,\, j \in \mathcal{I},\, i \neq j$ by expanding the matrix of explanatory variables with cross term $x_i x_j$. Let $g = \left|\mathcal{I}\right|$ denote the cardinality of $\mathcal{I}$, and let $h = \binom{g}{2}$ denote the number of combinations $i,\, j$, then the output of this step is the expanded matrix of explanatory variables $\boldsymbol{X}^* \in \mathbb{M}^{n \times (p+h)}$.

3. In the last step we again use the algorithm from §4 to find a linear regression model with the desirable properties for the data set $(\boldsymbol{y}, \boldsymbol{X}^*)$.

The output of this heuristic is the top three best linear regression models with cross terms included.

6.3 Computational Experiments

Data set

In this section we illustrate the performance of our heuristic based on a synthetic data set. We generated a synthetic data set using a similar procedure as in §5, for which we used the parameter settings as $n = 500$, $p = 100$, $\rho = 0$, $\Delta \boldsymbol{X} = 0$. Also, we consider again $k = 10$, so that $\beta_i = 1$ if $i \bmod p/k = 0$ and zero otherwise. Furthermore, we include one cross term $x_i x_j$ with regression coefficient $\beta = 1$ for $i \neq j$ and for which $i \bmod p/k = 0$ and $j \bmod p/k = 0$. That is, we include that one cross term $x_i x_j$, for some random i and j, that effects the response linearly with coefficient $\beta = 1$.

Results

In Table 10 we show the performance of our heuristic. We use the same structure in the table as in §5.1. We refer to p as the number of explanatory variables in the data without cross terms, because data naturally comes in the format without cross terms rather than with cross terms. We report the number of true nonzero regression coefficients as in §5.1, but note that we have, in contrast to §5.1, one extra nonzero regression coefficient, namely the cross term.

As in §5.1 we compare our approach with Lasso. For the Lasso computations we selected the cross terms to be considered similar as the heuristic. That is, we consider for Lasso the cross terms that may be selected by Lasso based on the nonzero regression coefficients in the first step. Both the heuristic and Lasso identify the true nonzero regression coefficients including the added cross term. Furthermore, Lasso brings in additional noise variables which is in line with the results found in §5.1. However, note that the number of additional noise variables brought in by Lasso is significantly larger than in §5.1. In contrast with our algorithm, Lasso produces models that are not interpretable because of the large number of additional noise variables. We find it notable that our heuristic truly identifies the nonzero cross term and does not bring in additional cross terms although the cross terms with zero coefficient have individual explanatory variables that have nonzero regression coefficient. Lastly, we note that the computational time is more than twice as large as in §5.1, which is because the heuristic uses the algorithm of §4 in the first step to find the relevant cross terms and in the last step uses the algorithm again with cross terms included. Therefore, it is expected that the computational time is approximately a factor two of the results in §5.1. We conclude that our heuristic produces significantly better models than Lasso, in the case where cross terms are included, because it results in more sparse linear regression models.

Table 10: Cross terms
$n = 500$, $p = 100$, $\rho = 0$, $\boldsymbol{\Delta X} = \boldsymbol{0}$.

σ^2	MIQO Γ	K^*	TP	R^2	MaxCor	Cond	Time	Lasso K^*	TP	R^2	MaxCor	Cond
0.5	0.107	11	11	0.941	0.130	1.579	9.478	124	11	0.940	0.203	11.601
1	0.094	11	11	0.888	0.124	1.594	9.512	130	11	0.893	0.276	9.657
2	0.091	11	11	0.802	0.128	1.645	9.455	147	11	0.814	0.280	9.184

7 Conclusion

We considered the linear regression model and defined the problem of finding a high quality linear regression model with the desirable properties of a linear regression model in a reasonable amount of time. The MIQO approach from Bertsimas and King (2015) is compared with the Lasso approach on both synthetic and real data sets to illustrate the performance of the MIQO approach on basic structure data set. Furthermore, we illustrated performance of the MIQO approach on a real data set with nonlinear transformations included and on a real data set with categorical variables. Lastly, we designed a synthetic data set in which we incorporated cross terms.

In §5.1 the performance of the algorithm is compared to the commonly used Lasso approach (Tibshirani (1996)) on basic structure data sets. Both the real as well as the synthetic data sets show that the algorithm produces linear regression models that achieve the desirable properties and are in general more sparse than the linear regression models resulting from the Lasso approach. The predictive power in terms of out-of-sample R^2 is comparable with Lasso. However, the algorithm produces linear regression models that are more interpretable than Lasso, because Lasso brings in additional noise variables compared to the algorithm. Furthermore, the algorithm avoids the issue of multicollinearity in contrast to the Lasso approach. Therefore, we conclude that the algorithm outperforms Lasso in terms of the desirable properties.

In §5.2 the performance of the algorithm is illustrated on two data sets with a special structure. For the data set with nonlinear transformations, we see that the algorithm choses nonlinear transformations of explanatory variables and has again comparable predictive power with Lasso. However, Lasso produces linear regression models with high pairwise multicollinearity, resulting in instable parameter estimates whereas the algorithm has low pairwise multicollinearity. For the data set with a group structure, we find that, although the algorithm chose to not include the grouped variables, the performance in terms of predictive power is comparable to group Lasso (Yuan and Lin (2006)). However, group Lasso produces linear regression models that are not particularly sparse in contrast to the algorithm. We conclude that the algorithm performs better on both special structure data sets in terms of interpretability and limited multicollinearity.

In §6 we incorporated cross terms and introduced a heuristic selection of cross terms. The performance of the heuristic is compared with Lasso in §6.3. Both the heuristic and Lasso selected the true nonzero regression coefficients including the cross term and have comparable predictive power in terms of out-of-sample R^2. However, Lasso again brings in additional noise variables in contrast to the heuristic, and therefore Lasso results in less interpretable models. The computation time of the heuristic is approximately doubled because of the incorporation of cross terms. We conclude that the heuristic performs better than Lasso in terms of interpretability when cross terms are incorporated.

In general we find that the algorithm produces high quality linear regression models that achieve the desirable properties of a linear regression model in a reasonable amount of time.

Bibliography

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

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

Bertsimas, D. and King, A. (2015). OR forum — An algorithmic approach to linear regression. *Operations Research*, 64(1):2–16.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Campbell, J. Y. and Thompson, S. B. (2007). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.

Chatterjee, S. and Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

DiCiccio, T. J. and Efron, B. (1996). Bootstrap confidence intervals. *Statistical science*, 11(3):189–212.

Efron, B. (1982). *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam.

Friedrich, R. J. (1982). In defense of multiplicative terms in multiple regression equations. *American Journal of Political Science*, 26(4):797–833.

Furnival, G. M. and Wilson, R. W. (1974). Regressions by leaps and bounds. *Technometrics*, 16(4):499–511.

Gurobi Optimization, I. (2016). Gurobi optimizer reference manual. `http://www.gurobi.com`.

Heij, C., De Boer, P., Franses, P. H., Kloek, T., Van Dijk, H. K., et al. (2004). *Econometric methods with applications in business and economics*. OUP Oxford.

MATLAB (2017). Version 9.2.0 (r2017b). `https://www.mathworks.com/`.

Miller, A. (2002). *Subset selection in regression*. CRC Press.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

RStudio Team (2015). *RStudio: Integrated Development Environment for R.* RStudio, Inc., Boston, MA. `http://www.rstudio.com/`.

Ryan, T. P. (2008). *Modern regression methods*, volume 655. John Wiley & Sons.

Tabachnick, B. G. and Fidell, L. S. (2001). *Using Multivariate Statistics.–4th.–Tabachnick and Fidell.* Allyn and Bacon.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

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

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

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.