https://doi.org/10.26219/heal.aueb.5951

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

DEPARTMENT OF STATISTICS

Thesis: Dantzig selector in linear models in n<p problems.

Postgraduate student

Name: Nikolaos

Surname: Georgopoulos

R. N.: 61523

Supervisor: Vassilis Vasdekis

ATHENS, FEBRUARY 2017

https://doi.org/10.26219/heal.aueb.5951

https://doi.org/10.26219/heal.aueb.5951

Thanksgivings

This work constitutes a thesis in the postgraduate program of Statistics in the University of Economics and Business in Athens. Before presenting the results of this thesis, I feel obliged to thank Mr Vassilis Vasdekis for the valuable guidance, the confidence as well as the assessment shown to me. The fruitful collaboration with him was of great value for the accomplishment of my thesis.

https://doi.org/10.26219/heal.aueb.5951

Contents

https://doi.org/10.26219/heal.aueb.5951

Chapter 1: Introduction

In many important statistical applications, the number of variables or parameters p is much larger than the number of observations n. In radiology and biomedical imaging, for instance, one is typically able to collect far fewer measurements about an image of interest than the unknown number of pixels. Examples in functional MRI and tomography all come to mind. High dimensional data frequently arise in genomics. Gene expression studies are a typical example: a relatively low number of observations (in the tens) is available, while the total number of genes assayed (and considered as possible regressors) is easily in the thousands. Other examples in statistical signal processing and nonparametric estimation include the recovery of a continuous-time curve or surface from a finite number of noisy samples.

From the above we understand that in many research fields, scientists work with data matrices with many variables p and comparably few observations n (large number of predictors relative to sample size, p>n). This causes serious problem in the estimation of the parameter $\beta \in R^p$ from the traditional linear regression model. The ordinary least squares (OLS) do not offer a good estimation. So many other approaches have been presented that outperform OLS. Most of them can be categorized into one of two groups. The first set of approaches uses some form of regularization on the regression coefficients to trade off increased bias for a possibly significant decrease in variance. While these approaches often produce improvements in prediction accuracy, the final fit may be difficult to interpret because all p variables will remain in the model. The second set of approaches begins by performing variable selection i.e. determining which $\beta_j \neq 0$. By implementing OLS on the reduced number of variables one can often gain both increased prediction accuracy as well as a more easily interpretable model.

Most recently interest has focused on an alternative class of methods which implement both the variable selection and the coefficient shrinkage in a single procedure. The methods we refer include the lasso (Robert Tibshirani, 1996), the adaptive lasso (Hui Zou, 2006), smoothly clipped absolute deviation (Jianqing Fan & Runze Li, 2001), the elastic net (Hui Zou & Trevor Hastie, 2005), the relaxed lasso

https://doi.org/10.26219/heal.aueb.5951

(Meinshausen, 2007), the Dantzig selector (Emmanuel Candès & Terence Tao, 2007), the variable inclusion and shrinkage algorithms (Peter Radchenko & Gareth M. James, 2008) and many variations of the above. Many of these methods and their variations use an L_1 penalty on the regression coefficients, which has the effect of shrinking some coefficients towards zero and setting others to exactly zero, thereby performing variable selection.

This thesis concerns the Dantzig selector, which is a method of dealing with the problem described above. The main scope is to define the Dantzig selector approach for linear models and then apply it to different data. In the second section, we will present the theoretical background of the method in our effort to figure out how it works. Simulated studies and real data examples will be given in the other two sections for better understanding of the method. The results will be studied in order to make useful comments and compare them with others that derive from analyses with different methods. Hence, we will be able to conclude in what cases Dantzig selector is preferable to other methods. Finally, we will summarize our work and we will discuss about this.

https://doi.org/10.26219/heal.aueb.5951

Chapter 2: Theoretical background of Dantzig selector and Lasso

Dantzig selector was presented by Emmanuel Candes and Terence Tao (2007) in their effort to estimate coefficients when the number of predictors is large relative to sample size, p>n. As we mentioned in the introduction, this section presents the theoretical background of our method. We know that there are several other methods that can help us cope with this problem. Some of these methods are lasso, visa, garotte etc. Lasso is the most prevalent method, so we are going to take a look at the theory of this method too.

2.1 The Dantzig selector

2.1.1 Definition

Beginning our presentation, we have to say that Dantzig selector works by minimizing the L_1 norm[1] of β subject to an L_∞ constraint on the correlation of the residuals with the predictors. Assume that we have the traditional linear regression model,

$$Y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i \qquad (i=1,\dots,n), \qquad (1)$$

where β is the vector of the coefficients,

p is the number of the parameters,

n is the number of the observations,

X is the matrix with the explanatory variables and

[1] Given a vector space *V* over a subfield *F* of the complex numbers, a norm on *V* is a function *p*: *V* → R with the following properties:
For all *a* ∈ *F* and all u, v ∈ *V*,

1. *p*(*a*v) = | *a* | *p*(v), (*absolute homogeneity* or *absolute scalability*).
2. *p*(u + v) ≤ *p*(u) + *p*(v) (*triangle inequality* or *subadditivity*).
3. If *p*(v) = 0 then v is the zero vector (*separates points*).

By the first axiom, absolute homogeneity, we have *p*(0) = 0 and *p*(−v) = *p*(v), so that by the triangle inequality *p*(v) ≥ 0 (*non-negativity*).
***p*-norm**
Main article: L^p space
Let *p* ≥ 1 be a real number.
$||x||_p := (\sum_{i=1}^{p} |x_i|^p)^{1/p}$
For *p* = 1 we get the taxicab norm, for *p* = 2 we get the Euclidean norm, and as *p* approaches ∞ the *p*-norm approaches the infinity norm or maximum norm. The *p*-norm is related to the generalized mean or power mean.

https://doi.org/10.26219/heal.aueb.5951

ε are the errors.

Let us also assume that in the above linear regression model the columns of the design matrix, X, have been normalized. The Dantzig selector estimator, $\hat{\beta}$, is the solution to the constrained optimization problem:

$$\min_{\tilde{\beta}\epsilon B} ||\tilde{\beta}||_1 \quad \text{subject to} \quad |X_j^T(Y\text{-}X\tilde{\beta})| \leq \lambda \quad (j = 1, \ldots, p), \qquad (2)$$

where $||.||_1$ represents the L_1 norm,

X_j is the jth column of the design matrix,

B represents the set of possible values for β, usually taken to be a subset of R^p and

λ is a tuning parameter.

Emmanuel Candes and Terence Tao (2007) propose to use as λ the quantity $\sigma\sqrt{2\log p}$. Specifically, they mention the following theorem:

THEOREM: Suppose $\beta\epsilon\boldsymbol{R}^p$ is any S-sparse vector of parameters obeying $\delta_{2S} + \vartheta_{S,2S} < 1$. Choose $\lambda_p = \sqrt{2logp}$ in (2) ($\lambda = \lambda_p \cdot \sigma$). Then with large probability, $\hat{\beta}$ obeys

$$||\hat{\beta} - \beta||_2^2 \leq C_1^2 \cdot (2logp) \cdot S \cdot \sigma^2,$$

with $C_1 = 4/(1 - \delta_S - \vartheta_{S,2S})$. Hence, for small values of $\delta_S + \vartheta_{S,2S}$, $C_1 \approx 4$. For concreteness, if one chooses $\lambda_p := \sqrt{2(1+a)logp}$ for each $\alpha \geq 0$, the bound holds with probability exceeding $1 - (\sqrt{\pi logp} \cdot p^{\alpha})^{-1}$ with the proviso that λ_p^2 substitutes 2logp in (2) ($\lambda = \lambda_p \cdot \sigma$).

For more details and the proof of the theorem see Candes and Tao (2007).
For the definition of quantities δ and θ see Candes and Tao (2005).

The L_1 minimization produces coefficient estimators that are exactly zero and hence can be used as a variable selection tool.

In this set-up X_j is assumed to have norm one, which is rarely the case in practice. However, this difficulty is easily resolved by reparameterizing the traditional linear regression model such that X_js have norm one. Let $d_j = ||X_j||_2$. Then computing $\hat{\beta}$ for the standardized X_js and retuning $\hat{\beta}_j/d_j$ produces the Dantzig selector estimator for β_j on the original scale.

https://doi.org/10.26219/heal.aueb.5951

For Gaussian error terms, (2) can be rewritten as

$$\min_{\tilde{\beta}\epsilon B} ||\tilde{\beta}||_1 \quad \text{subject to} \quad |l'_j(\tilde{\beta})| \leq \lambda/\sigma \quad (j = 1, \ldots, p), \qquad (3)$$

where l'_j is the partial derivative of the loglikelihood function with respect to β_j and $\sigma^2=var(\varepsilon_i)$. Hence, for $\lambda=0$, (3) will return the maximum likelihood estimator. Alternatively, the constraint in (2) is equivalent to $X^TY = X^TX\hat{\beta}$, which is simply the least squares normal equation. For $\lambda>0$ the Dantzig selector searches for the $\tilde{\beta}$ with the smallest L_1 norm that, within a fixed tolerance, satisfies the normal equations characterizing the maximum likelihood solution, i.e. the sparsest $\tilde{\beta}$ that is still reasonably consistent with the observed data. Even for p>n, in which case the likelihood equation has infinitely many solutions, this approach can still hope to identify the correct solution, provided β is sparse, because it attempts only to locate the sparsest $\tilde{\beta}$ close to the peak of the likelihood function. One might imagine that minimizing the L_0 norm, which counts the number of nonzero components of a vector, would be more appropriate than the L_1 norm. However, directly minimizing the L_0 norm is computationally intractable and one can show that, under suitable conditions, the L_1 norm will also provide a sparse solution.

The Dantzig selector has some nice properties. The first is that (2) can be formulated as a linear programming problem, so standard software can easily be used. The second useful property is theoretical. Emmanuel Candes and Terence Tao (2007) prove a tight non-asymptotic bound on the error in the estimator for β that is within a factor of log*p* of the error rate achieved if the true predictors are assumed known. Since log*p* grows very slowly, the Dantzig selector only pays a small price for adaptively choosing the significant variables.

Emmanuel Candes and Terence Tao (2007) also study the reason why one would want to constrain the size of the correlated residual vector X*r rather than the size of the residual vector r, where $r=Y-X\tilde{\beta}$. To explain it, we can suppose that an orthonormal transformation is applied to the data, giving Y'=UY, where U*U is the identity. Clearly, a good estimation procedure for estimating β should not depend upon U (after all, one could apply U* to return to the original problem). It turns out that the estimation procedure (2) is actually invariant with respect to orthonormal transformations applied to the data vector since the feasible region is invariant:

https://doi.org/10.26219/heal.aueb.5951

$(UX)^T(UX\tilde{\beta}-UY)=X^*(X\tilde{\beta}-Y)$. In contrast, had we defined the feasibility region with $\sup_i|r_i|$ being smaller than a fixed threshold, then the estimation procedure would not be invariant. There are other reasons aside from this. One of them is that we would obviously want to include in the model explanatory variables that are highly correlated with the data Y. Consider the situation in which a residual vector is equal to a column X_i of the design matrix X. Suppose, for simplicity, that the components of X_i all have about the same size. That is, about $1/\sqrt{n}$, and assume that σ is slightly larger than $1/\sqrt{n}$. Had we used a constraint of the form $\sup_i|r_i|\leq\lambda_n\sigma$ (with perhaps λ_n of size about $\sqrt{2logn}$), the vector of residuals would be feasible, which does not make any sense. In contrast, such a residual vector would not be feasible for (2) for reasonable values of the noise level, and the ith variable would be rightly included in the model.

2.1.2 Variance estimation

Another fundamental problem in statistical modelling is variance estimation. In ultrahigh dimensional linear regression where the dimensionality is much larger than the sample size, traditional variance estimation techniques are not applicable. Recent advances in variable selection in ultrahigh dimensional regression make this problem accessible. One of the major problems in ultrahigh dimensional regression is the high spurious correlation between the unobserved realized noise and some of the predictors. As a result, the realized noises are actually predicted when extra irrelevant variables are selected, leading to a serious underestimate of the level of noise.

A key assumption which makes the high dimensional problems solvable is the sparsity condition: the number of non-zero components is small compared with the sample size. With sparsity, variable selection can identify the subset of important predictors and improve the model interpretability and predictability. There are several methods for variance estimation in the setting of an ultrahigh dimensional linear model. Both the asymptotic theory and the empirical result show that the RCV (Refitted cross-validation) estimator, which is presented by Jianqing Fan, Shaojun

https://doi.org/10.26219/heal.aueb.5951

Guo and Ning Hao (2012), is the best among all estimators, so we are going to present it.

At first, we describe a simple approach to the problem and then we explain the reason why we use RCV to improve the above estimator. This procedure has two stages. In the first stage, a model selection tool is applied to select a model which, if is not exactly the true model, includes all important variables with moderate model size (smaller than the sample size). In the second stage, the variance is estimated by an ordinary least squares method based on the variables selected in the first stage. Obviously, this method works well if we can recover exactly the true model in the first stage. This is usually difficult to achieve in ultrahigh dimensional problems.

Unfortunately, this naïve two-step approach can seriously underestimate the level of noise even with the sure screening property in the first stage owing to spurious correlation that is inherent in ultrahigh dimensional problems. When the number of irrelevant variables is huge, some of these variables have large sample correlations with the realized noises. Hence, almost all variable selection procedures will, with high probability, select those spurious variables in the model when the model is overfitted, and the realized noises are actually predicted by several spurious variables, leading to a serious underestimate of the residual variance.

In order to remove the influence of spurious variables in the second stage, we present the RCV (Refitted cross-validation) method. This method requires only that the model selection procedure in the first stage has a sure screening property. The idea is as follows. We assume that the sample size n is even for simplicity and split randomly the sample into two groups. In the first stage, an ultrahigh dimensional variable selection method is applied to these two data sets separately, which yields two small sets of selected variables. In the second stage, the ordinary least squares method is used to re-estimate the coefficient β and variance σ^2. Differently from the naïve two-stage method, we apply ordinary least squares again to the first subset of the data with the variables selected by the second subset of the data and vice versa. Taking the average of these two estimators, we obtain our estimator of σ^2. The refitting in the second stage is fundamental to reduce the influence of the spurious variables in the first stage of variable selection.

https://doi.org/10.26219/heal.aueb.5951

For better understanding we quote the implementation of the above idea of RCV as presented by Jianqing Fan, Shaojun Guo and Ning Hao (2012). Consider a data set with sample size n, which is randomly split into two even data sets $(Y^{(1)}, X^{(1)})$ and $(Y^{(2)}, X^{(2)})$. First, a variable selection tool is performed on $(Y^{(1)}, X^{(1)})$ and let M_1 denote the set of variables selected. The variance σ^2 is then estimated on the second data set $(Y^{(2)}, X^{(2)})$, namely

$$\hat{\sigma}_1^2 = (Y^{(2)T}(I_{n/2} - P_{M1}^{(2)})Y^{(2)}) / (n/2-|M_1|),$$

where $P_{M1}^{(2)} = X_{M1}^{(2)}(X_{M1}^{(2)T}X_{M1}^{(2)})^{-1}X_{M1}^{(2)T}$. Similarly, we use the second data set $(Y^{(2)}, X^{(2)})$ to select the set of important variables M_2 and the first data set $(Y^{(1)}, X_{M2}^{(1)})$ for estimation of σ^2, resulting in

$$\hat{\sigma}_2^2 = (Y^{(1)T}(I_{n/2} - P_{M2}^{(1)})Y^{(1)}) / (n/2-|M_2|).$$

We define the final estimator as

$$\hat{\sigma}^2_{RCV} = (\hat{\sigma}_1^2 + \hat{\sigma}_2^2)/2.$$

An alternative is the weighted average defined by

$$\hat{\sigma}^2_{WRCV} = (Y^{(2)T}(I_{n/2} - P_{M1}^{(2)})Y^{(2)} + Y^{(1)T}(I_{n/2} - P_{M2}^{(1)})Y^{(1)}) / (n-|M_1|-|M_2|).$$

When $|M_1| = |M_2|$, we have $\hat{\sigma}^2_{RCV} = \hat{\sigma}^2_{WRCV}$.

In this procedure, although M_1 includes some extra unimportant variables besides the important variables, these extra variables will play minor roles when we estimate σ^2 by using the second data set along with refitting since they are just some random unrelated variables over the second data set. Furthermore, even when some important variables are missed in the first stage of model selection, they have a good chance of being well approximated by the other variables selected in the first stage to reduce modelling biases. Thanks to the refitting in the second stage, the best linear approximation of those selected variables is used to reduce the biases. Therefore, a larger selected model size gives us, not only a better chance of sure screening, but also a way to reduce modelling biases in the second stage when some important variables are missing. This explains why the RCV method is insensitive to the model size selected.

https://doi.org/10.26219/heal.aueb.5951

There are two natural extensions of the aforementioned RCV technique (Jianqing Fan, Shaojun Guo and Ning Hao, 2012):

(a) K-fold data splitting: the first natural extension is to use a K-fold data splitting technique rather than twofold splitting. We can divide the data into K groups and select the model with all groups except one, which is used to estimate the variance with refitting. We may improve the sure screening probability with this K-fold method since there are now more data in the first stage. However, there are only n/K data points in the second stage for refitting. This means that the number of variables that are selected in the first stage should be much less than n/K. This makes the ability of sure screening difficult in the first stage. For this reason we prefer working on the twofold RCV.

(b) Repeated data splitting: there are many ways to split the data randomly. Hence, many RCV variance estimators can be obtained. We may take the average of the resulting estimators.This reduces the influence of the randomness in the data splitting.

Now we have to say that many statistical problems require knowledge of the residual variance, especially for high or ultrahigh dimensional linear regression. Here we quote a couple of applications: (a) Constructing confidence intervals for coefficients, (b) Genomewide association studies and (c) Model selection. We are going to briefly outline the first one, which is the only that we studied for this thesis.

So we could say that a natural application is to use estimated $\hat{\sigma}_{RCV}$ to construct confidence intervals for non-vanishing estimating coefficients. Let $\hat{\beta}_M$ be the Dantzig selector estimator, with the corresponding design matrix X_M. Then for each j∈ M[2], the 1-α confidence interval for β_j is

$$\hat{\beta}_j \pm z_{1-\alpha/2} c_j \hat{\sigma}_{RCV},$$

in which c_j is the diagonal element of the matrix $(X_M{}^T X_M)^{-1}$ that corresponds to the jth variable. Our simulation studies show that a confidence interval is accurate and haw a similar performance to the case where σ is known.

The confidence intervals can also be constructed on the basis of the raw materials in the RCV. For example, for each element in $M=M_1 \cap M_2$, we can take the

[2] $M=M_1 \cap M_2$

https://doi.org/10.26219/heal.aueb.5951

average of the refitted coefficients as the estimate of the regression coefficients in the set M, and $(S_1 + S_2)\hat{\sigma}^2_{RCV}/4$ as the corresponding estimated covariance matrix, where $S_1 = (X_M^{(1)T}X_M^{(1)})^{-1}$ is computed on the basis of the first half of the data at the refitting stage and $S_2 = (X_M^{(2)T}X_M^{(2)})^{-1}$ is computed on the basis of the second half of the data.

Some may wonder why we use the RCV method for estimating the variance since the dimensions of the problem have been reduced. The reason is why we want to attenuate the influence of spurious variables that are entered into the selected model and to improve the accuracy of estimation. The corrections of biases due to spurious correlation are dramatic. The essential difference between this approach and the naive two-stage approach is that the regression coefficients in the first stage are discarded and refitted by using the second half of the data and hence the spurious correlations in the first stage are significantly reduced at the second stage. The variance estimation is unbiased as long as the models selected in the first stage contain all relevant variables, namely they have a sure screening property. It turns out that this simple RCV method improves dramatically the performance of the naive two-stage procedure. Clearly, the RCV can also be used to do model selection itself, reducing the influence of spurious variables.

To appreciate why, suppose that a predictor has a big sample correlation with the response (realized noise in the null model) over the first half of the data set and is selected into the model by a model selection procedure. Since the two halves of the data set and is selected into the model by a model selection procedure. Since the two halves of the data set are independent and a chance that a given predictor is highly correlated with realized noise is small, it is very unlikely that this predictor has a large sample correlation with the realized noise over the second half of the data set. Hence, its influence on the variance estimation is very small when refitted and estimating the variance over the second half will not cause any bias. This argument is also true for the non-null models provided that the model selected includes all important variables (Jianqing Fan, Shaojun Guo and Ning Hao, 2012).

https://doi.org/10.26219/heal.aueb.5951

2.2 The Lasso

2.2.1 Definition

As we mentioned above, Lasso is one of the most prevalent methods for dealing with methods that ols estimators are unable to yield reliable results. This method also shrinks some coefficients and sets others to 0. Let's see the theory in which the operation of this procedure is based on, as presented by Robert Tibshirani (1996).

Suppose that we have data (x^i, y_i), i=1,2,….,N, where $x^i=(x_{i1}, …., x_{ip})^T$ are the predictor variables and y_i are the responses. As in the usual regression set-up, we assume either that the observations are independent or that the y_is are conditionally independent given the x_{ij}s. We assume that the x_{ij} are standardized so that $\sum_i x_{ij}/N = 0$, $\sum_i x_{ij}^2/N = 1$.

Letting $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)^T$, the lasso estimate $(\hat{\alpha}^3, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \{ \sum_{i=1}^{N} (y_I - \alpha - \sum_j \beta_j x_{ij})^2 \} \quad \text{subject to } \sum_j |\beta_j| \leq t.$$

Here $t \geq 0$ is a tuning parameter. Now, for all t, the solution for α is $\hat{\alpha} = \bar{y}$. We can assume without loss of generality that $\bar{y}=0$ and hence omit α.

Computation of the solution to equation above is a quadratic programming problem with linear inequality constraints. The parameter $t \geq 0$ controls the amount of shrinkage that is applied to the estimates. Let $\hat{\beta}_j^0$ be the full least squares estimates and let $t_0=\sum |\hat{\beta}_j^0|$. Values of $t<t_0$ will cause shrinkage of the solutions towards 0, and some coefficients may be exactly equal to 0. For example, if $t=t_0/2$, the effect will be roughly similar to finding the best subset of size p/2. Note also that the design matrix need not be of full rank.

2.2.2 Standard errors

Since the lasso estimate is a non-linear and non-differentiable function of the response values even for a fixed value of t, it is difficult to obtain an accurate estimate of its standard error. One approach is via the bootstrap: either t can be fixed or we may optimize over t for each bootstrap sample. Fixing t is analogous to

[3] α corresponds to the constant.

https://doi.org/10.26219/heal.aueb.5951

selecting a best subset, and then using the least squares standard error for that subset.

An approximate closed form estimate may be derived by writing the penalty$\sum|\beta_j|$ as $\sum\beta_j^2/|\beta_j|$. Hence, at the lasso estimate $\tilde{\beta}$, we may approximate the solution by a ridge regression of the form $\beta^*=(X^TX+\lambda W^-)^{-1}X^Ty$ where W is a diagonal matrix with diagonal elements $|\tilde{\beta}_j|$, W^- denotes the generalized inverse of W and λ is chosen so that $\sum|\beta_j|^*=t$. The covariance matrix of the estimates may then be approximated by

$$(X^TX+\lambda W^-)^{-1}X^TX(X^TX+\lambda X^-)^{-1}\hat{\sigma}^2,$$

where $\hat{\sigma}^2$ is an estimate of the error variance. A difficulty with this formula is that it gives an estimated variance of 0 for predictors with $\hat{\beta}_j=0$.

This approximation also suggests an iterated ridge regression algorithm for computing the lasso estimate itself, but this turns to be quite inefficient. However, it is useful for selection of the lasso parameter t (Robert Tibshirani, 1996).

https://doi.org/10.26219/heal.aueb.5951

Chapter 3: Algorithms used – Simple problem

In this section we describe the steps we followed to study Dantzig selector. At first, we have to say that we used the programming language R to encode our simulation schemes[4]. So we quote the commands we used to the code, the algorithms we used and a simple simulation scheme that helped us to make useful conclusions. We explain the reasons why we used this simulation scheme and we present the results that it yields.

3.1 Commands and algorithms used

3.1.1 The Dantzig selector

The basic command we use to apply Dantzig selector to our data is the *nloptr* which is an R interface to NLopt, a free/open-source library for nonlinear optimization, providing a common interface for a number of different free optimization routines (manual of programming language R, package 'nloptr'). NLopt addresses general nonlinear optimization problems of the form:

$\min f(x), x \text{ in } R^n$

$s.t.\ g(x) \leq 0$

$h(x) = 0$

$lb \leq x \leq ub$

where f is the objective function to be minimized and x represents the n optimization parameters. This problem may optionally be subject to the bound constraints (also called box constraints), lb and ub. For partially or totally unconstrained problems the bounds can take -Inf or Inf. One may also optionally have m nonlinear inequality constraints (sometimes called a nonlinear programming problem), which can be specified in g(x), and equality constraints that can be specified in h(x). Note that not all of the algorithms in NLopt can handle constraints.

In our case the objective function is $\min_{\tilde{\beta} \in B} ||\tilde{\beta}||_1$ with the constraint $|X_j^T(Y - X\tilde{\beta})| \leq \lambda$ (j=1, …., p) and the vector of coefficients we want to estimate is β.

[4] The code used is shown in the appendix.

https://doi.org/10.26219/heal.aueb.5951

The arguments of the command *nloptr* concern the starting values, the objective function, the constraint, the lower bound, the upper bound and the algorithm that is used for the optimization. The objective function, the constraint and the bounds are mentioned in the previous paragraph, so we are going to say some words about the starting values and the algorithm that is used.

As we design the simulation schemes, we know the true values of the parameters we want to estimate. So we can use as starting values, the true values of the parameters. Hence, we present results that refer to the true values of the parameters, but we also comment on what happens when we use random numbers as starting values.

Concerning the algorithm that is used we have to say that there are plenty of algorithms that can be used. COBYLA is the one we choose to work with and the reason is that it is a nonlinear derivative-free constrained optimization that uses a linear approximation approach (M.J.D. Powell, 2007). The algorithm is a sequential trust-region algorithm (originally with a monotonically decreasing radius ρ of a spheric trust region) that employs linear approximations to the objective and constraint functions, where the approximations are formed by linear interpolation at n+1 points in the space of the variables and tries to maintain a regular-shaped simplex over the iterations. A small modification was made to the original algorithm, that permits an increase of the trust-region radius ρ in special situations. A sequence of iterations is performed with a constant trust-region radius ρ until the computed objective function reduction is much less than the predicted reduction. Then, the trust region radius ρ is reduced. The trust-region radius is increased only if the computed function reduction is relatively close to the predicted reduction and the simplex is well-shaped. The start radius ρ_{beg} and the final radius ρ_{end} can be specified. The convergence to small values of ρ_{end} (high precision) may take many calls of the function and constraint modules and may result in slow convergence (M.J.D. Powell, 2007).

https://doi.org/10.26219/heal.aueb.5951

3.1.2 The Lasso

For calling the Lasso we use the "glmnet" package. The command *cv.glmnet* from this package, helps us to extract easily the values of the tuning parameter λ and the estimates of the coefficients that corresponds to each λ. The reasons for using this package are that it is more friendly, it has a wider selection of function and it enable us to use every λ we want and see the corresponding results.

Our procedure was run with two different values of the LASSO tuning parameter λ, each obtained by using a 10-fold CV. The first parameter, termed λ_{min}, minimizes the CV error. The second parameter, termed $\lambda_{1\text{se}}$, is a stronger penalty to guard against model overfit and is the maximum value with its CV error within one standard error of the minimum CV error (manual of programming language R, package 'glmnet').

3.2 Simulation scheme - Simple problem for examining the appropriateness of the method

Firstly, we present a very simple simulation scheme which we used in order to ascertain if our method works well. We generate randomly three variables from multivariate normal distribution with mean 5 and covariance matrix $\begin{matrix} 4 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 4 \end{matrix}$, which means that the correlation between the variables is r=0.5. Subsequently, we describe a procedure that is repeated 200 times in order to observe more reliable results. At each time of this procedure we produce data (Y) using the first two of the three aforementioned variables and residuals (ε). The residuals derive from a normal distribution with mean 0 and standard deviation $\sqrt{6}$ and they differ for each one of the 200 times that the procedure is repeated. We also choose the coefficients that are used to produce the data and we assume that the number of observations is n=100.

https://doi.org/10.26219/heal.aueb.5951

Our aim is to use Dantzig selector including the third variable and observe the estimates of the coefficients. We would like to see estimations close to the actual coefficients. Especially, for the third variable we would like to see an estimation of the coefficient close to zero, so that this variable should not be included in the model.

The problem we described above is very simple, so we can use the method of the ordinary least squares for the estimation of coefficients. By using this method we are able to compare the results arising from both methods and decide if Dantzig selector works well. The table below sums up our results:

True coefficients	OLS		Dantzig selector		
	Estimate	Standard errors	Estimate	Standard errors	RMSE
13	13.07	0.78	13.03	0.70	0.77
5	5.01	0.16	5.01	0.14	0.15
7	6.99	0.16	6.99	0.13	0.14
0	-0.01	0.15	-0.01	-[5]	0.24

Table 3.1: Table with the expected values of the estimates, the variances and the MSE. It contains the results for both OLS and Dantzig selector. It also contains the true coefficients in order to compare the methods with the reality.

Observing the table, we could say that Dantzig selector seems to works pretty well at this simple problem. It has almost the same results as the OLS method and the variance of each coefficient is almost the same. We could also say that the arising estimates are unbiased since the RMSE does not seem to differ significantly with the standard errors of the coefficients. We also have to point out that Dantzig selector yields good results even if we use random numbers as starting values for the coefficients. Equally good results are observed in cases when we change the correlation between the variables or when we change the value of λ, which is the parameter included in the constraint. Except the case of medium correlation (r=0.5), we try zero correlation (r=0) and correlation close to one (r=0.96). We also use 3 different values of λ: i) λ=0,001, ii) $\lambda=\sigma\sqrt{2logp}$ (this λ is proposed by Emmanuel Candes and Terence Tao, 2007) and iii) $\lambda\cong 2\sigma\sqrt{2logp}$.

[5] In Dantzig selector, variance is computed only for the coefficients included in the model.

https://doi.org/10.26219/heal.aueb.5951

As we mentioned in Section 2, we can use variance estimation to construct confidence intervals for coefficients. Following the procedure we described in Section 2, we constructed confidence intervals for the first three coefficients (the constant and the coefficients of the first two variables) in order to check, in an alternative way, if Dantzig selector operates well. We repeated the procedure 200 times and we measured how many times the estimation of each coefficient was inside the corresponding confidence interval. We expect a percentage about 95%. Results ensure that our method operates pretty well. The estimation of the constant seems to “fall” inside the confidence interval at the 95,5% of the times. The estimation of the coefficient of the first variable is consistent with the corresponding confidence interval at a percentage of 97%. Finally, the estimation of the coefficient of the second variable seems to be in harmony with the confidence interval at the 96% of the times measured. The confidence intervals of the estimators that derive from the ols method are 94,5%, 95%, 97,5% and 95% respectively.

All the above results ensure us that Dantzig selector implementation is correct and it can be used to simple problems. So, now we are going to use this formula in more complicated problems than the above. We are trying to study in which cases is appropriate to use Dantzig selector. These complicated simulation schemes are presented in the next section.

https://doi.org/10.26219/heal.aueb.5951

Chapter 4: More complicated simulation schemes

In this section we present more complicated simulation schemes than the one presented in the previous section. We try to compare Dantzig selector with Lasso by using both methods at the same datasets for different number of predictors and observations each time. Finally, we comment on the results and we conclude in which cases Dantzig selector can be used trustworthy and also when it is preferable to Lasso.

These simulation schemes refer to different combinations of the number of predictors (p) and the number of observations (n). We increase the number of predictors and decrease the number of observations in order to result in n<<p, which is the case in study. We work as in the simple problem presented in the previous section. We use only a few of the variables to produce data and subsequently, we use our methods with all the variables to see if true variables are selected for the final model. We expect zero estimations of the coefficients of the variables that do not participate in the production of the data.

For every combination of the above we try different values of λ and different correlations[6] between the variables. We observe the "behavior" of Dantzig selector and we try to decide in which cases we can use it with reliable results. We also use the method of Lasso for all of the aforementioned combinations and we compare the results arising from both methods. For each case, we run an algorithm 500 times and we present the expectation value of the estimates and the root mean square error (RMSE[7]) of every coefficient[8]. We also mention the running time of each algorithm (in seconds). These measures are going to help us in our effort to compare the two methods. So let's study each one of the cases separately by observing the tables and figures[9] that are presented below:

[6] The way we define correlations between the variables is presented at the appendix.

[7] RMSE is given by $\sqrt{(\frac{1}{B}\sum_{i=1}^{B}(\hat{\theta}_i-\theta)^2)}$, where θ is the true value of the parameter, $\hat{\theta}_i$ is the estimation of the parameter for each one of the repeats and B is the number of repeats, here B=500.

[8] We explain at the appendix the reason why we do not quote the variance.

[9] We present these figures for better understanding of how estimations are affected when λ, n and r change.

https://doi.org/10.26219/heal.aueb.5951

4.1: 10 predictors – 8 zero coefficients

In this case only two of the predictors are used to produce our data. Hence, we expect from Dantzig selector and Lasso to detect which are these two predictors. We study this problem for different number of observations and different values of correlation between the variables.

- n=100 and r=0

Dantzig selector

	λ=0.001 (running time=108 s)		λ=$\sigma\sqrt{2logp}$=3.46 (running time=72 s)		λ=7 (running time=49 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.93	2.33	9.22	4.38	5.66	7.70
5	5.01	0.15	5.07	0.15	5.13	0.19
7	7.00	0.15	7.09	0.18	7.18	0.24
0	0.00	0.12	0.07	0.13	0.12	0.17
0	0.00	0.13	0.07	0.15	0.15	0.20
0	0.00	0.11	0.03	0.10	0.07	0.13
0	0.01	0.13	0.08	0.15	0.14	0.19
0	0.01	0.14	0.11	0.18	0.21	0.25
0	0.01	0.14	0.11	0.18	0.20	0.24
0	0.00	0.12	0.06	0.15	0.14	0.19
0	0.00	0.12	0.06	0.14	0.12	0.16

Table 4.1.1: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

The above table shows us that if we have 100 observations and the variables are independent we have good estimations, close to the real ones, when λ=0.001. Estimations seem to be good for the other values of λ too. We detect only one problem at the estimation of the constant. Specifically, when we increase λ, the estimation of the constant "breaks down". The best RMSE seems to correspond at the case of λ=0.001. This can be obtained if we take a look at the figure below:

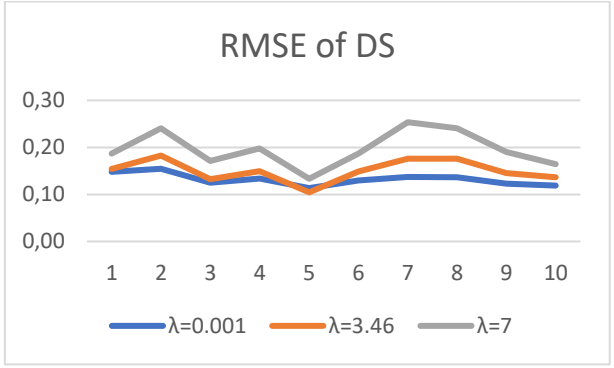

Figure 4.1.1: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

https://doi.org/10.26219/heal.aueb.5951

Lasso (running time=32 seconds)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.20	1.94	16.52	3.71
5	4.88	0.19	4.64	0.39
7	6.88	0.20	6.63	0.40
0	0.00	0.06	0.00	0.01
0	0.00	0.07	0.00	0.02
0	0.00	0.06	0.00	0.01
0	0.00	0.07	0.00	0.02
0	0.00	0.07	0.00	0.02
0	0.00	0.07	0.00	0.01
0	0.00	0.07	0.00	0.01
0	0.00	0.06	0.00	0.02

Table 4.1.2: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

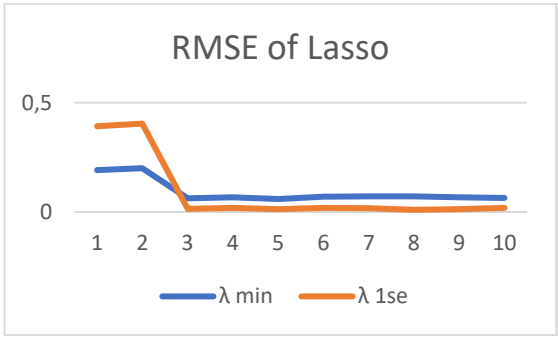

Figure 4.1.2: Figure which shows the RMSE for each value of λ (Lasso).

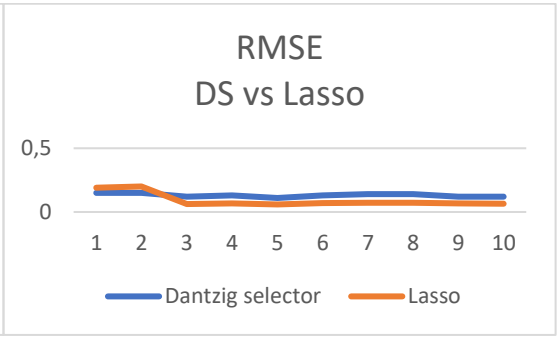

Figure 4.1.3: Figure which compares the RMSEs Between the two methods (Dantzig selector & Lasso).

Observing the results for the two different cases of Lasso, we conclude that λ min has better estimations and better RMSE from λ 1se for the current combination of n and r.

Now, if we would like to make a conclusion about which method is better, we could say that Dantzig selector has more accurate estimations when λ=0.001 but Lasso has better RMSE and that is obvious if we look figure 4.1.3.

- n=100 and r=0.5

Only in this case, we also present the standard errors of the coefficients in order to compare them with the corresponding RMSEs and decide if estimates are unbiased or no. We used the method presented in Chapter 2 (2.1.2) for computing the standard errors.

https://doi.org/10.26219/heal.aueb.5951

Dantzig selector

	λ=0.001 (running time=115 s)			λ=σ$\sqrt{2logp}$=3.46 (running time=97 s)			λ=7 (running time=93 s)		
True coefficients	Estimates	RMSE	Standard errors	Estimates	RMSE	Standard errors	Estimates	RMSE	Standard errors
13	12.94	0.94	0.84	12.49	1.07	0.89	11.84	1.49	0.68
5	5.00	0.19	0.17	4.99	0.17	0.17	4.98	0.18	0.11
7	7.00	0.18	0.14	6.99	0.18	0.15	6.94	0.20	0.12
0	0.01	0.16	-	0.00	0.15	-	0.02	0.14	-
0	-0.01	0.16	-	-0.02	0.15	-	-0.03	0.14	-
0	-0.01	0.17	-	-0.01	0.16	-	0.00	0.14	-
0	0.00	0.16	-	0.04	0.17	-	0.07	0.18	-
0	0.00	0.17	-	0.00	0.16	-	0.02	0.15	-
0	0.01	0.16	-	0.02	0.15	-	0.01	0.13	-
0	0.01	0.18	-	0.05	0.19	-	0.13	0.22	-
0	-0.01	0.19	-[10]	0.03	0.19	-	0.06	0.19	-

Table 4.1.3: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison. It is the only table which contains the standard errors of the coefficients

For every λ, the estimators are very good. The best RMSE seems to correspond at the case of λ=3.46, but we should say that RMSE has fallen for all the values of λ. These conclusions become more comprehensible if we look the figure below and compare it with figure 4.1.1.

Comparing the standard errors with the corresponding RMSEs, we conclude that they are close together, so the estimators are almost unbiased. The case of λ= σ$\sqrt{2logp}$=3.46 offers the best estimators.

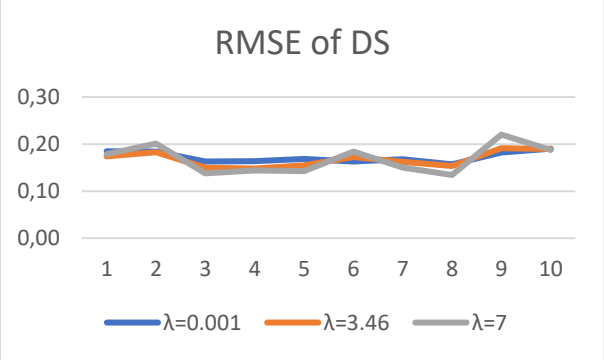

Figure 4.1.4: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

[10] Standard errors are computed only for the coefficients of the variables that are included in the model.

https://doi.org/10.26219/heal.aueb.5951

Lasso (running time=33 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	13.33	0.89	15.47	2.62
5	4.88	0.19	4.73	0.31
7	6.88	0.21	6.73	0.31
0	0.01	0.08	0.00	0.00
0	0.02	0.08	0.01	0.03
0	0.02	0.08	0.01	0.03
0	0.03	0.09	0.01	0.04
0	0.02	0.10	0.01	0.04
0	0.02	0.08	0.01	0.03
0	0.02	0.11	0.00	0.03
0	0.03	0.10	0.01	0.04

Table 4.1.4: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso continues to have lower RMSE from Dantzig selector only in the case of λ min. λ 1se gives high RMSE, while the estimations are good enough for both λ.

Figures 4.1.5 and 4.1.6 help us to make the above conclusions.

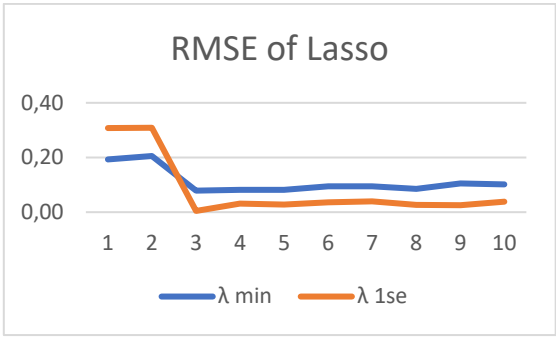

Figure 4.1.5: Figure which shows the RMSE for each value of λ (Lasso).

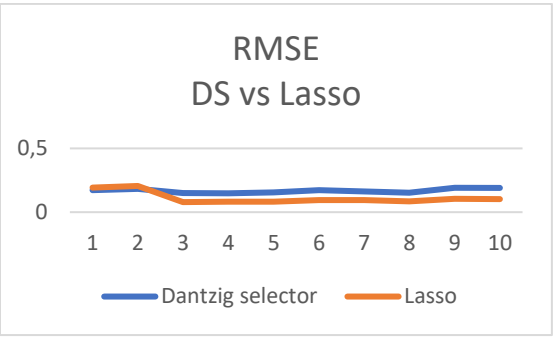

Figure 4.1.6: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=100 and r=0.96

Dantzig selector

	λ=0.001 (running time=137 s)		λ=$\sigma\sqrt{2logp}$=3.46 (running time=128 s)		λ=7 (running time=106 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.99	0.31	12.95	0.30	12.90	0.32
5	5.01	0.22	4.98	0.21	4.95	0.20
7	7.01	0.18	6.99	0.17	6.97	0.17
0	0.00	0.19	0.00	0.16	0.00	0.15
0	0.01	0.18	0.01	0.17	0.01	0.15
0	0.00	0.17	0.01	0.17	0.02	0.15

https://doi.org/10.26219/heal.aueb.5951

0	-0.01	0.18	0.00	0.17	0.01	0.14
0	0.00	0.17	0.00	0.16	0.01	0.14
0	-0.01	0.17	0.01	0.16	0.02	0.15
0	-0.01	0.21	0.00	0.19	-0.02	0.16
0	0.01	0.18	0.01	0.15	0.02	0.15

Table 4.1.5: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Excellent results are observed for all the values of λ when correlation is r=0.96. Estimations are almost accurate and the RMSE is almost the same for all values of λ. The claim for the RMSE is more comprehensible if we take a look at the figure below:

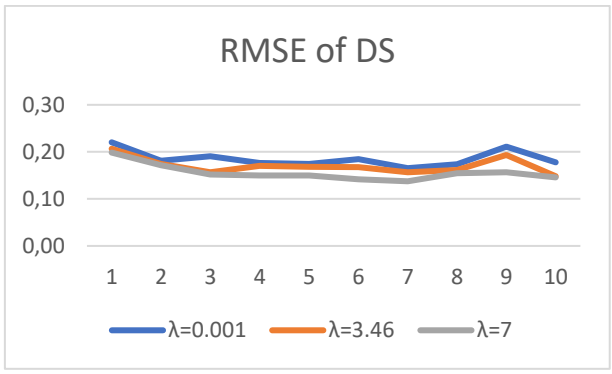

Figure 4.1.7: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

Lasso (running time=32 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.01	1.06	14.16	1.20
5	4.87	0.19	4.86	0.20
7	6.82	0.23	6.80	0.24
0	0.01	0.04	0.01	0.04
0	0.01	0.04	0.01	0.04
0	0.01	0.05	0.01	0.05
0	0.01	0.03	0.01	0.03
0	0.01	0.02	0.00	0.02
0	0.01	0.04	0.01	0.04
0	0.01	0.03	0.01	0.03
0	0.01	0.04	0.01	0.03

Table 4.1.6: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=100 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

λ min and λ 1se have almost the same RMSE. Lasso seems to lag behind the Dantzig selector in the case of high correlation when we talk about the estimations

https://doi.org/10.26219/heal.aueb.5951

of non-zero coefficients. RMSE for the rest of the coefficients are pretty good. Let's take a look at the two figures below to understand it in a better way:

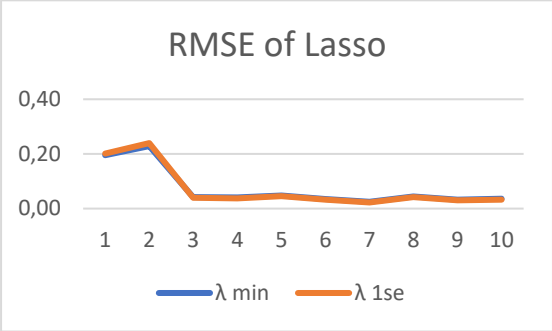

Figure 4.1.8: Figure which shows the RMSE for each value of λ (Lasso).

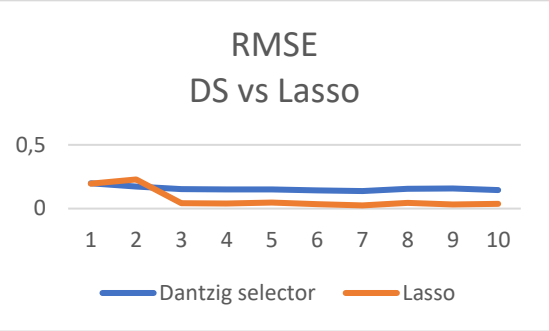

Figure 4.1.9: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=20 and r=0

Dantzig selector

	λ=0.001 (running time=231 s)		λ=$\sigma\sqrt{2logp}$=3.46 (running time=46 s)		λ=7 (running time=32 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	10.69	6.95	0.00	13.00	0.00	13.00
5	5.05	0.37	5.29	0.41	5.34	0.43
7	7.06	0.33	7.47	0.53	7.50	0.55
0	-0.01	0.46	0.00	0.39	0.11	0.30
0	-0.01	0.46	-0.18	0.41	-0.02	0.29
0	0.09	0.35	0.44	0.49	0.46	0.50
0	0.08	0.59	0.27	0.47	0.30	0.38
0	0.04	0.36	0.16	0.31	-0.07	0.22
0	0.10	0.43	0.48	0.57	0.36	0.43
0	0.07	0.45	0.33	0.48	0.25	0.37
0	0.09	0.50	0.44	0.58	0.32	0.43

Table 4.1.7: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

For the same number of predictors (p), we decreased the number of observations (n) and that seems to affect the RMSE for each λ. We observe large values of the RMSE, especially for the constant. Except the constant, estimations continue to be good enough. Figure 4.1.10, which is presented below, can be very useful:

https://doi.org/10.26219/heal.aueb.5951

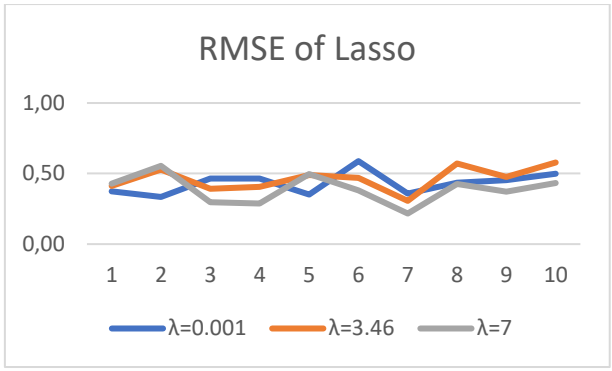

Figure 4.1.10: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

Lasso (running time=32 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	15.83	5.16	18.46	6.39
5	4.76	0.38	4.47	0.61
7	6.74	0.43	6.40	0.70
0	-0.02	0.22	-0.01	0.09
0	0.02	0.20	0.01	0.09
0	-0.06	0.19	-0.04	0.13
0	0.00	0.27	0.00	0.13
0	0.00	0.15	0.00	0.07
0	-0.03	0.20	-0.01	0.09
0	-0.01	0.24	0.01	0.12
0	-0.01	0.27	-0.01	0.14

Table 4.1.8: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso exhibits the same "behavior" as the Dantzig selector when we decrease the number of observations (n). RMSE has "broken down" and estimations are less accurate than before. Better values of the RMSE are given by Lasso. The following figures helped us to make these comments.

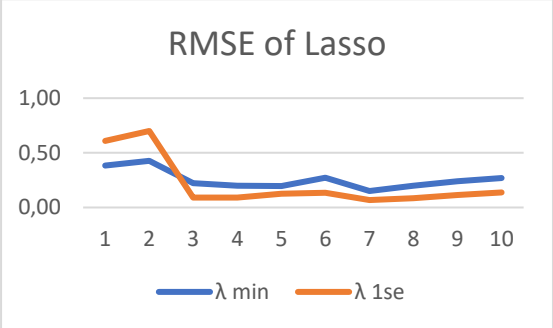

Figure 4.1.11: Figure which shows the RMSE for each value of λ (Lasso).

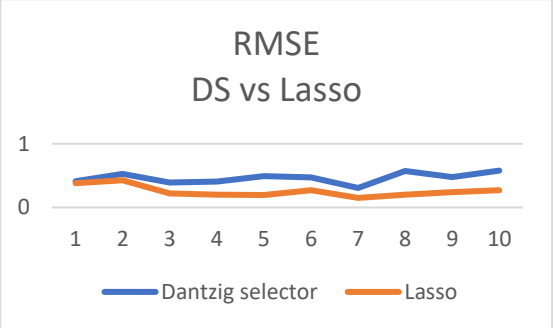

Figure 4.1.12: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

https://doi.org/10.26219/heal.aueb.5951

- n=20 and r=0.5

Dantzig selector

	λ=0.001 (running time=223 s)		$\lambda=\sigma\sqrt{2logp}$=3.46 (running time=67 s)		λ=7 (running time=42 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.89	3.29	6.22	7.45	0.97	12.15
5	5.02	0.54	5.47	0.68	5.91	1.02
7	7.01	0.41	7.37	0.51	7.56	0.65
0	-0.01	0.53	0.07	0.39	0.11	0.27
0	-0.01	0.53	-0.14	0.33	-0.18	0.33
0	0.03	0.60	-0.28	0.51	-0.12	0.42
0	0.02	0.46	0.17	0.38	0.12	0.29
0	-0.04	0.82	0.09	0.33	0.25	0.37
0	0.02	0.64	0.79	0.93	1.17	1.24
0	-0.03	0.43	-0.01	0.32	-0.15	0.33
0	0.03	0.49	-0.22	0.42	-0.30	0.41

Table 4.1.9: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

An increase in the correlation improves the estimations, especially for the case of λ=0.001, in which we observe very good estimates and a much better RMSE than before. The other two values of λ do not offer a good estimation of the constant.

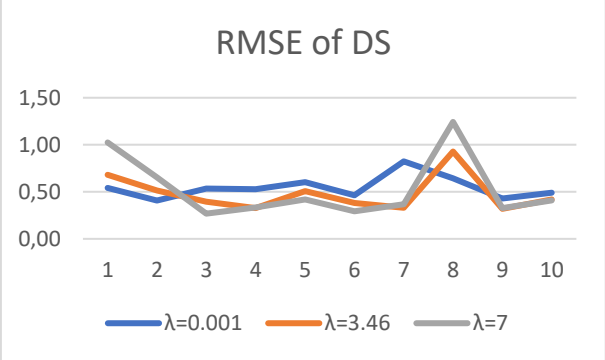

Figure 4.1.13: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

Lasso (running time=32 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.60	3.18	17.71	5.57
5	4.66	0.56	4.32	0.81
7	6.75	0.44	6.46	0.66
0	0.05	0.26	0.03	0.16
0	0.05	0.24	0.03	0.13
0	0.03	0.31	0.02	0.14

https://doi.org/10.26219/heal.aueb.5951

0	0.05	0.26	0.03	0.14
0	-0.01	0.35	0.00	0.18
0	-0.01	0.35	0.00	0.17
0	0.04	0.21	0.03	0.12
0	0.04	0.23	0.04	0.14

Table 4.1.10: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Improvement is also observed at Lasso, especially for λ min. Dantzig selector has lower RMSE for the non-zero coefficients, while Lasso has better RMSE for the remaining ones. In other words, Dantzig selector offers better estimations of the variables with non-zero coefficients, while Lasso detects more effectively the variables that we do not need.

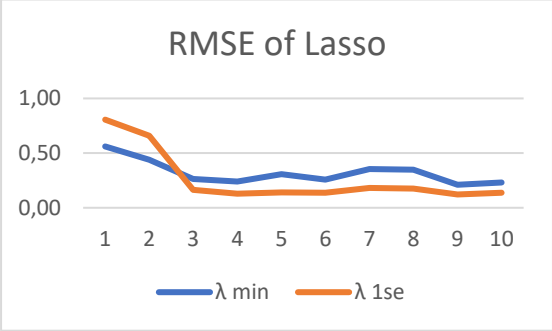

Figure 4.1.14: Figure which shows the RMSE for each value of λ (Lasso).

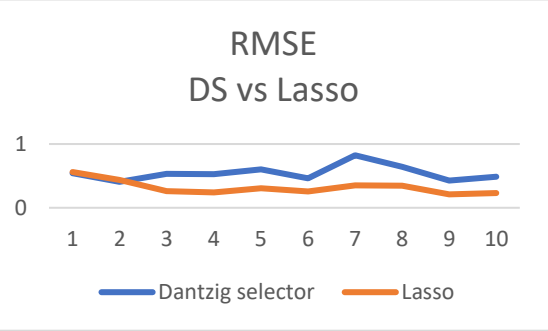

Figure 4.1.15: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=20 and r=0.96

Dantzig selector

	λ=0.001 (running time=371 s)		λ=$\sigma\sqrt{2logp}$=3.46 (running time=123 s)		λ=7 (running time=82 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.96	0.92	12.54	0.93	12.26	1.05
5	4.97	0.46	4.95	0.40	4.84	0.40
7	7.00	0.45	6.96	0.37	7.01	0.32
0	-0.01	0.60	0.07	0.38	0.05	0.25
0	-0.03	0.83	-0.10	0.35	-0.02	0.19
0	0.00	0.49	0.13	0.38	0.14	0.28
0	0.02	0.49	0.05	0.31	0.05	0.21
0	-0.01	0.57	-0.03	0.35	-0.01	0.23
0	0.01	0.38	0.06	0.31	0.09	0.26
0	0.04	0.62	0.03	0.37	0.02	0.23
0	0.00	0.49	-0.08	0.34	-0.12	0.27

Table 4.1.11: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

https://doi.org/10.26219/heal.aueb.5951

Putting correlation close to one results in having surprisingly good estimations for all values of λ. RMSE is better for the cases of λ=3.49 and λ=7 and that is also seen in the figure below:

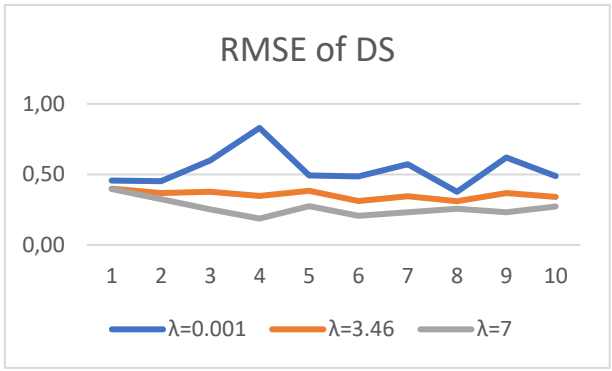

Figure 4.1.16: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

Lasso (running time=30 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.57	1.71	15.00	2.11
5	4.72	0.40	4.69	0.42
7	6.70	0.42	6.67	0.44
0	0.05	0.13	0.04	0.13
0	0.01	0.06	0.01	0.06
0	0.03	0.10	0.02	0.10
0	0.08	0.18	0.07	0.16
0	0.02	0.08	0.02	0.07
0	0.03	0.09	0.03	0.09
0	0.05	0.14	0.04	0.13
0	0.04	0.11	0.03	0.10

Table 4.1.12: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=20 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Almost the same results are observed for the two values of λ. Comparing the RMSEs between the two methods, we could say that Dantzig selector estimates more effectively only the non-zero coefficients. The above results are easily seen in the figures 4.1.17 and 4.1.18:

https://doi.org/10.26219/heal.aueb.5951

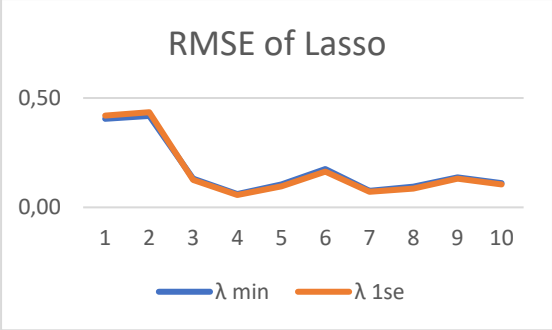

Figure 4.1.17: Figure which shows the RMSE for each value of λ (Lasso).

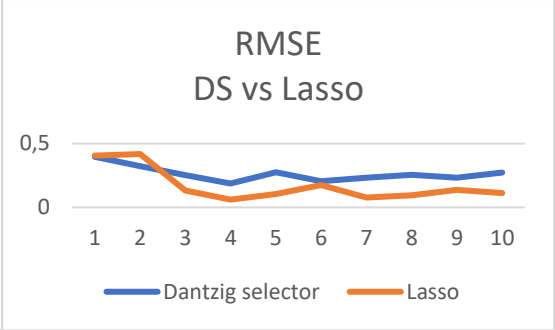

Figure 4.1.18: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=7 and r=0

Dantzig selector

	λ=0.001 (running time=37 s)		$\lambda=\sigma\sqrt{2logp}$=3.46 (running time=8 s)		λ=7 (running time=6 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	2.76	11.13	0.00	13.00	0.00	13.00
5	5.27	0.48	5.34	0.48	5.40	0.51
7	7.03	1.05	7.53	0.82	7.71	0.92
0	0.01	0.40	0.13	0.26	0.19	0.27
0	0.28	0.46	0.37	0.47	0.33	0.41
0	-0.06	0.39	0.07	0.26	0.10	0.22
0	0.00	0.42	0.10	0.33	0.10	0.25
0	0.37	0.54	0.24	0.37	0.22	0.32
0	0.19	0.54	0.24	0.45	0.20	0.35
0	0.50	0.76	0.30	0.45	0.21	0.33
0	0.37	0.60	0.23	0.37	0.14	0.25

Table 4.1.13: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

In this case we have less observations (n) than the predictors (p). For r=0, constant cannot be estimated effectively. The RMSEs are close together for all values of λ.

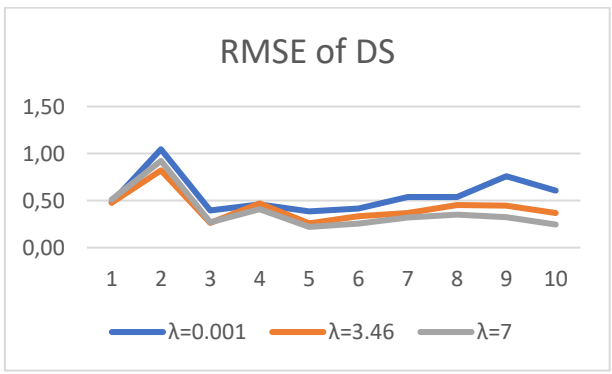

Figure 4.1.19: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

https://doi.org/10.26219/heal.aueb.5951

Lasso (running time=26 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	72.28	59.50	74.78	61.81
5	1.00	4.05	0.18	15.30
7	0.03	6.97	0.00	7.00
0	0.01	0.15	0.00	0.01
0	-1.00	1.19	-0.11	0.33
0	0.74	1.03	0.06	0.29
0	0.00	0.03	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-0.06	0.21	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

Table 4.1.14: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso offers very bad estimations of non-zero coefficients. On the other hand 5/8 zero coefficients are estimated exactly and the corresponding RMSEs are close to zero which is pretty good.

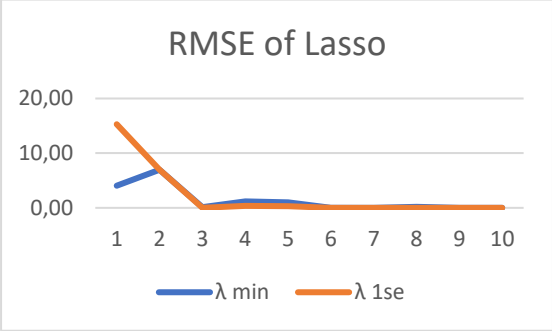

Figure 4.1.20: Figure which shows the RMSE for each value of λ (Lasso).

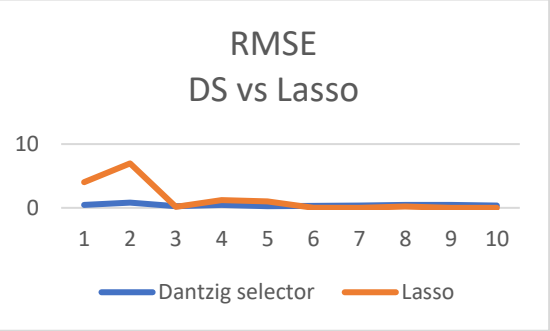

Figure 4.1.21: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=7 and r=0.5

Dantzig selector

	λ=0.001 (running time=38 s)		$\lambda=\sigma\sqrt{2logp}$=3.46 (running time=31 s)		λ=7 (running time=12 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.26	2.92	0.23	12.83	0.00	13.00
5	4.96	0.62	4.50	1.06	4.71	0.97
7	6.85	1.02	8.85	2.14	7.92	1.30
0	-0.07	0.29	-0.33	0.57	-0.27	0.42
0	-0.01	0.96	-0.32	0.54	-0.04	0.22
0	0.16	0.48	0.21	0.35	0.32	0.44
0	0.04	0.48	-1.41	1.64	-0.48	0.80

https://doi.org/10.26219/heal.aueb.5951

0	-0.05	0.25	1.38	1.53	0.96	1.14
0	0.12	0.54	0.28	0.44	0.36	0.47
0	0.17	0.56	0.27	0.48	0.09	0.33
0	-0.04	0.47	0.57	0.71	0.50	0.66

Table 4.1.15: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

An increase in the correlation improves the estimations when λ=0.001. Bad estimations and large RMSEs are observed for the values of λ=3.46 and λ=7. For better understanding, we look figure 4.1.22:

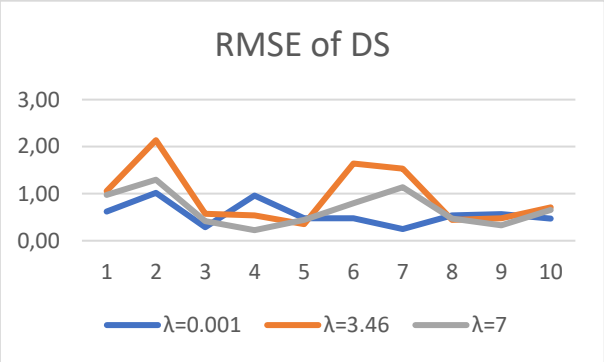

Figure 4.1.22: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

Lasso (running time=27 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.47	2.76	18.30	6.27
5	4.69	0.67	4.42	0.84
7	6.47	0.90	6.36	0.93
0	0.01	0.07	0.00	0.03
0	0.08	0.21	0.04	0.15
0	0.07	0.21	0.03	0.12
0	0.09	0.25	0.06	0.19
0	0.05	0.17	0.03	0.13
0	0.06	0.18	0.01	0.09
0	0.16	0.41	0.05	0.22
0	0.03	0.10	0.01	0.06

Table 4.1.16: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Improvement is observed at the estimation of non-zero coefficients, but this does not happen to the remaining. The RMSE of the zero coefficients seems to increase. We would prefer Dantzig selector for estimating the non-zero coefficients, but Lasso for the rest.

https://doi.org/10.26219/heal.aueb.5951

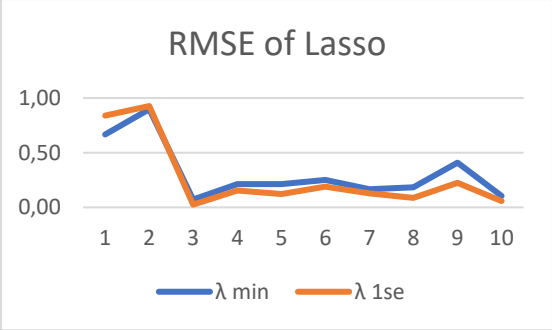

Figure 4.1.23: Figure which shows the RMSE for each value of λ (Lasso).

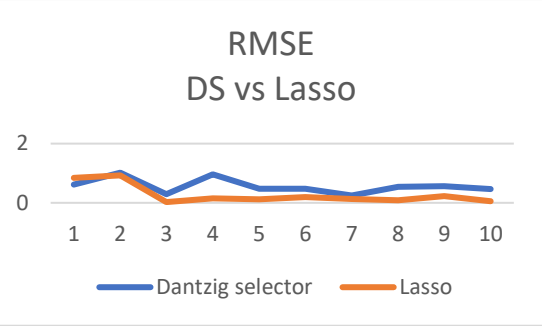

Figure 4.1.24: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

- n=7 and r=0.96

Dantzig selector

	λ=0.001 (running time=116 s)		λ=$\sigma\sqrt{2logp}$=3.46 (running time=72 s)		λ=7 (running time=67 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	12.97	1.83	10.39	3.08	9.09	4.27
5	5.12	1.34	2.86	2.62	1.60	3.69
7	6.87	1.33	6.70	1.05	7.48	1.08
0	0.12	1.32	0.02	0.10	0.01	0.06
0	-0.04	1.27	0.03	0.09	0.02	0.06
0	0.03	0.51	0.57	0.76	0.91	1.04
0	-0.10	0.83	-0.22	0.38	-0.17	0.27
0	-0.11	0.45	0.17	0.32	0.18	0.30
0	-0.05	0.38	0.98	1.22	1.31	1.47
0	0.10	1.10	0.48	0.73	0.42	0.59
0	0.06	0.61	0.21	0.40	0.25	0.39

Table 4.1.17: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

We have good estimates only for the case of λ=0.001, but the RMSE has increased for each one of the coefficients for every λ. It is easily seen if we compare figure 4.1.25 with figures 4.1.19 and 4.1.22.

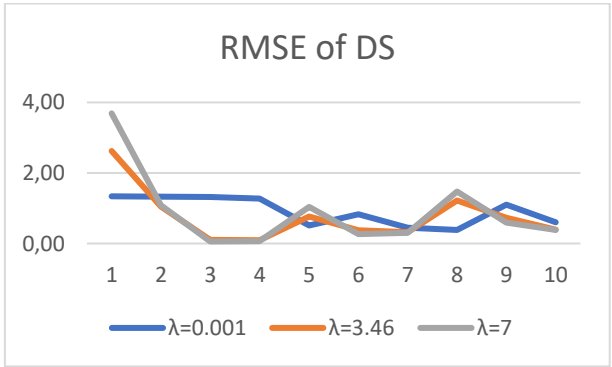

Figure 4.1.25: Figure which shows the RMSE for each predictor and for each value of λ (Dantzig selector).

https://doi.org/10.26219/heal.aueb.5951

Lasso (running time=27 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	14.15	1.94	14.84	2.39
5	4.16	1.36	4.14	1.36
7	6.00	1.51	5.92	1.57
0	0.01	0.03	0.00	0.01
0	0.52	1.02	0.54	1.04
0	0.10	0.29	0.08	0.24
0	0.01	0.03	0.00	0.01
0	0.03	0.17	0.03	0.14
0	0.65	0.17	0.65	1.17
0	0.14	0.40	0.13	0.38
0	0.01	0.05	0.01	0.04

Table 4.1.18: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=7 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso gives better results than the case r=0.5 but the RMSE is not good enough for all the coefficients. λ min and λ 1se give the same RMSEs. For n=7 and r=0.96, the two methods give almost the same RMSEs.

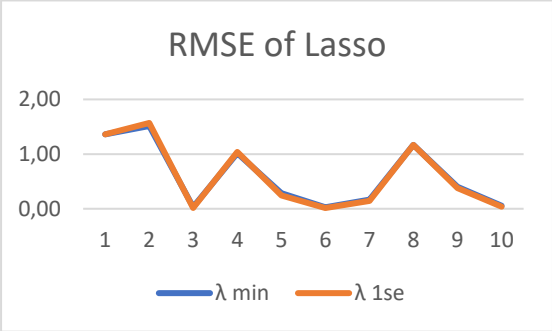

Figure 4.1.26: Figure which shows the RMSE for each value of λ (Lasso).

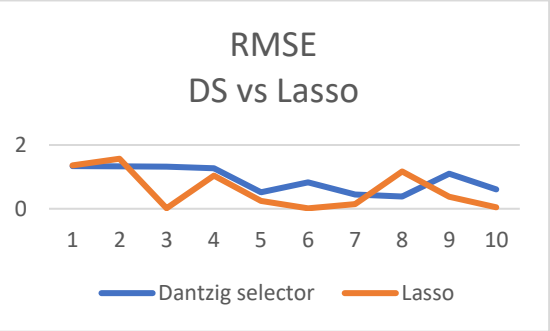

Figure 4.1.27: Figure which compares the RMSEs between the two methods (Dantzig selector & Lasso).

Generally, we could say that adding correlation between the variables causes improvement to our estimations. We have more accurate estimates and better RMSEs when we have high correlations. We should also mention that decreasing the number of observations complicates the algorithm. For small number of observations we have not accurate estimates. We could also say that λ plays an important role at the arising estimations. In general, λ that gives the best RMSE seems to be λ= $\sigma\sqrt{2logp}$, which is the one suggested by Candes and Tao. Although all these comments refer to the case of known starting values of the coefficients, we

https://doi.org/10.26219/heal.aueb.5951

have to say that if we use random numbers as starting values in Dantzig selector, results do not seem to change dramatically.

Before studying the next case, we should mention some observations about the running time of the algorithm. Specifically, we observe decrease in the running time of the algorithm when we increase λ. We also observe increase in the running time when we add correlation between the variables. Algorithm needs more time to give results if the correlation is high. Finally, we mention that Lasso is faster than Dantzig selector.

https://doi.org/10.26219/heal.aueb.5951

4.2: 100 predictors – 90 zero coefficients

In this case we use only 10 predictors to produce our data, so we expect from Dantzig selector and Lasso to detect which are the 90 predictors with zero coefficients. We do not present figures for this case because we have large number of predictors (p) and figures are not articulate, so they do not help us to make conclusions.

- n=25 and r=0

Dantzig selector

	λ=0.001 (running time=5385 s)		$\lambda=\sigma\sqrt{2logp}$=4.9 (running time=14700 s)		λ=10 (running time=16080 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	-1.03	14.05	0.00	13.00	0.00	13.00
5	3.82	1.34	3.80	1.35	3.84	1.34
7	7.11	0.38	7.07	0.36	7.05	0.36
8	9.30	1.41	9.25	1.35	9.20	1.33
6	7.26	1.42	7.29	1.46	7.36	1.53
9	8.75	0.44	8.77	0.46	8.87	0.39
5	5.12	0.41	5.06	0.40	5.01	0.40
7	1.13	5.89	1.03	5.98	0.91	6.11
7	7.33	0.78	7.29	0.79	7.14	0.74
6	7.06	1.14	7.02	1.09	7.00	1.07
4	2.53	1.49	2.52	1.53	2.57	1.49
0	0.62	0.63	0.61	0.63	0.59	0.49
0	0.28	0.30	0.26	0.29	0.24	0.27
0	0.24	0.27	0.26	0.29	0.29	0.31
0	0.29	0.32	0.29	0.31	0.29	0.31
0	-0.07	0.14	-0.07	0.13	-0.06	0.12
0	0.01	0.11	0.03	0.11	0.03	0.08
0	0.14	0.18	0.14	0.19	0.12	0.16
0	0.16	0.21	0.17	0.21	0.18	0.21
0	0.29	0.30	0.28	0.30	0.28	0.31
0	0.13	0.18	0.15	0.18	0.16	0.19
0	0.02	0.12	0.03	0.11	0.03	0.12
0	0.14	0.20	0.15	0.20	0.14	0.18
0	-0.08	0.12	-0.07	0.10	-0.06	0.11
0	0.06	0.13	0.05	0.12	0.05	0.11
0	0.15	0.19	0.15	0.18	0.16	0.19
0	-0.05	0.13	-0.04	0.10	-0.01	0.11
0	-0.09	0.15	-0.07	0.11	-0.05	0.11
0	-0.01	0.11	0.00	0.10	0.01	0.10
0	0.10	0.16	0.10	0.15	0.13	0.17
0	0.11	0.15	0.12	0.15	0.13	0.17

https://doi.org/10.26219/heal.aueb.5951

0	0.09	0.16	0.07	0.13	0.07	0.13
0	0.13	0.18	0.14	0.18	0.15	0.20
0	0.02	0.10	0.03	0.09	0.04	0.11
0	-0.04	0.15	-0.03	0.16	-0.01	0.14
0	0.43	0.45	0.41	0.43	0.38	0.39
0	0.36	0.38	0.34	0.36	0.33	0.34
0	-0.01	0.12	-0.02	0.11	-0.02	0.09
0	-0.11	0.17	-0.10	0.17	-0.09	0.16
0	0.21	0.24	0.16	0.20	0.16	0.19
0	0.10	0.17	0.12	0.17	0.10	0.15
0	0.19	0.23	0.19	0.22	0.22	0.26
0	0.05	0.14	0.05	0.12	0.05	0.11
0	-0.11	0.15	-0.11	0.15	-0.09	0.14
0	0.11	0.17	0.09	0.14	0.07	0.13
0	0.25	0.27	0.24	0.27	0.22	0.25
0	0.13	0.16	0.14	0.19	0.13	0.17
0	0.13	0.17	0.13	0.18	0.11	0.16
0	-0.09	0.13	-0.09	0.13	-0.07	0.11
0	0.04	0.12	0.04	0.12	0.04	0.10
0	0.05	0.13	0.04	0.11	0.04	0.12
0	0.33	0.38	0.31	0.37	0.30	0.37
0	-0.23	0.28	-0.23	0.27	-0.20	0.25
0	-0.02	0.14	-0.02	0.12	-0.02	0.10
0	0.22	0.25	0.19	0.22	0.19	0.22
0	0.21	0.25	0.22	0.25	0.21	0.23
0	-0.27	0.30	-0.28	0.31	-0.27	0.30
0	0.31	0.33	0.31	0.33	0.30	0.32
0	-0.04	0.12	-0.02	0.10	0.00	0.09
0	0.14	0.19	0.13	0.17	0.13	0.17
0	0.08	0.14	0.06	0.12	0.04	0.11
0	0.37	0.38	0.38	0.39	0.38	0.40
0	0.26	0.28	0.26	0.30	0.24	0.26
0	-0.14	0.18	-0.13	0.17	-0.13	0.18
0	0.36	0.42	0.33	0.38	0.32	0.38
0	0.31	0.34	0.30	0.31	0.27	0.30
0	-0.06	0.12	-0.05	0.11	-0.04	0.09
0	0.11	0.18	0.13	0.19	0.13	0.17
0	0.11	0.19	0.09	0.17	0.08	0.16
0	0.04	0.12	0.07	0.13	0.09	0.15
0	0.12	0.16	0.13	0.17	0.12	0.16
0	0.01	0.12	0.01	0.10	0.02	0.11
0	0.17	0.20	0.17	0.21	0.18	0.21
0	0.03	0.13	0.03	0.11	0.04	0.10
0	-0.09	0.18	-0.07	0.17	-0.07	0.17
0	0.07	0.12	0.06	0.12	0.06	0.11
0	-0.24	0.26	-0.23	0.25	-0.20	0.22
0	-0.07	0.13	-0.05	0.11	-0.03	0.10
0	-0.12	0.17	-0.10	0.16	-0.09	0.14
0	-0.08	0.16	-0.05	0.15	-0.03	0.12
0	0.06	0.11	0.06	0.12	0.07	0.11

https://doi.org/10.26219/heal.aueb.5951

0	0.27	0.30	0.26	0.29	0.25	0.28
0	0.02	0.12	0.02	0.10	0.03	0.10
0	-0.20	0.24	-0.21	0.24	-0.18	0.22
0	0.17	0.23	0.16	0.23	0.15	0.22
0	0.02	0.12	0.02	0.11	0.01	0.09
0	-0.12	0.17	-0.07	0.14	-0.05	0.11
0	-0.07	0.15	-0.05	0.14	-0.02	0.15
0	0.10	0.15	0.10	0.15	0.11	0.14
0	0.14	0.18	0.15	0.18	0.16	0.19
0	0.28	0.30	0.26	0.29	0.23	0.25
0	0.05	0.12	0.04	0.12	0.04	0.12
0	0.04	0.13	0.05	0.14	0.07	0.14
0	0.24	0.26	0.22	0.24	0.21	0.24
0	0.06	0.12	0.06	0.12	0.05	0.10
0	0.19	0.22	0.16	0.20	0.13	0.18
0	0.08	0.14	0.06	0.12	0.05	0.12
0	-0.06	0.14	-0.05	0.12	-0.04	0.10
0	0.08	0.14	0.11	0.16	0.12	0.19
0	0.10	0.14	0.10	0.15	0.09	0.13
0	-0.14	0.18	-0.15	0.19	-0.15	0.18

Table 4.2.1: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

The number of predictors (p=100) is large relative to the number of observations (n=25), which is the case in study. Coefficients are not estimated effectively, and the values of the RMSE are large, especially for non-zero coefficients. The good news is that the rest of the coefficients have estimations close to zero and small values of the RMSE.

Lasso (running time=41 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	318.69	305.73	339.71	326.73
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
8	4.59	3.44	0.58	7.44
6	0.00	6.00	0.00	6.00
9	0.00	9.00	0.00	9.00
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
7	0.00	7.00	0.00	7.00
6	0.00	6.00	0.00	6.00
4	0.00	4.00	0.00	4.00
0	0.00	0.01	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

0	0.00	0.00	0.00	0.00
0	0.00	0.04	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.03	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.06	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.02	0.22	0.00	0.00
0	-0.01	0.19	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.03	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.01	0.18	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.04	0.00	0.00
0	0.00	0.10	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.07	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.05	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.08	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-0.01	0.16	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.06	0.00	0.00
0	0.00	0.05	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.05	0.00	0.00
0	0.00	0.05	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.26	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.15	0.00	0.00
0	0.01	0.14	0.00	0.00
0	0.01	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.02	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.10	0.30	0.00	0.00

Table 4.2.2: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso sets to zero almost all the coefficients. Large values observed for the RMSEs of non-zero coefficients, while the RMSEs of the remaining coefficients is close to zero.

None of the methods is reliable in this case. They have great difficulty in estimating properly the coefficients when the number of observations is n=25 and the variables are independent.

https://doi.org/10.26219/heal.aueb.5951

- n=25 and r=0.5

Dantzig selector

	λ=0.001 (running time=6180 s)		$\lambda=\sigma\sqrt{2logp}$=4.9 (running time=24000 s)		λ=10 (running time=18600 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	8.28	5.23	0.03	12.97	0.04	12.97
5	5.86	1.07	5.61	0.85	4.72	0.64
7	7.14	0.58	6.74	0.55	7.65	0.88
8	7.42	0.77	7.42	0.74	7.28	0.93
6	6.60	0.86	6.01	0.56	6.04	0.56
9	7.06	2.08	6.26	2.81	8.19	1.06
5	5.42	0.90	5.33	0.78	3.79	1.32
7	0.02	7.00	0.06	6.94	0.8	6.22
7	8.21	1.38	8.23	1.37	4.87	2.19
6	6.09	0.66	5.25	0.94	6.63	0.78
4	6.01	2.14	6.42	2.51	5.14	1.26
0	1.12	1.13	1.46	1.47	0.9	0.90
0	-0.27	0.36	-0.19	0.26	0.14	0.20
0	-0.29	0.35	-0.15	0.23	0.24	0.34
0	-0.17	0.21	-0.17	0.22	0.39	0.44
0	0.02	0.13	0.15	0.18	-0.12	0.19
0	0.01	0.14	0.05	0.11	0.15	0.23
0	0.16	0.28	0.17	0.27	0.02	0.09
0	0.06	0.16	0.13	0.19	0.08	0.20
0	0.02	0.20	0.04	0.15	0.21	0.34
0	0.10	0.27	0.05	0.17	-0.01	0.04
0	0.05	0.23	0.03	0.15	0.62	0.66
0	0.01	0.14	0.04	0.11	0.04	0.09
0	-0.12	0.19	-0.05	0.10	0.00	0.10
0	0.01	0.22	0.10	0.18	0.39	0.42
0	0.16	0.27	0.18	0.27	0.01	0.06
0	0.28	0.36	0.35	0.41	0.35	0.39
0	-0.23	0.26	-0.08	0.12	-0.09	0.13
0	0.14	0.23	0.10	0.19	0.09	0.21
0	-0.15	0.26	-0.09	0.19	0.59	0.61
0	-0.14	0.22	0.07	0.14	0.07	0.18
0	0.15	0.23	0.16	0.22	0.08	0.14
0	-0.01	0.15	0.04	0.10	0.09	0.15
0	0.35	0.42	0.59	0.63	-0.01	0.08
0	0.22	0.30	0.14	0.21	0.03	0.08
0	0.03	0.14	0.11	0.16	0.05	0.14
0	-0.10	0.22	0.02	0.14	0.21	0.25
0	0.12	0.19	0.12	0.16	0.32	0.34
0	0.12	0.24	0.05	0.16	0.05	0.14
0	0.04	0.23	0.02	0.15	0.12	0.20
0	0.20	0.25	0.26	0.31	0.07	0.13
0	0.12	0.19	0.06	0.13	-0.13	0.20

https://doi.org/10.26219/heal.aueb.5951

0	0.16	0.22	0.17	0.22	0.03	0.07
0	0.19	0.23	0.16	0.20	-0.10	0.19
0	-0.07	0.17	0.05	0.14	0.06	0.10
0	0.23	0.28	0.13	0.20	0.18	0.22
0	0.14	0.26	0.07	0.16	0.08	0.20
0	0.07	0.20	0.10	0.18	0.55	0.60
0	0.57	0.58	0.75	0.77	0.68	0.71
0	-0.33	0.37	-0.34	0.39	0.06	0.14
0	-0.05	0.21	0.00	0.12	0.08	0.16
0	0.05	0.14	0.07	0.11	0.15	0.21
0	-0.10	0.24	-0.05	0.18	0.24	0.27
0	0.11	0.22	0.12	0.20	-0.15	0.20
0	0.02	0.14	0.09	0.14	0.04	0.14
0	0.60	0.63	0.70	0.73	0.32	0.40
0	0.37	0.40	0.47	0.49	0.01	0.03
0	-0.27	0.30	-0.31	0.34	0.13	0.18
0	-0.15	0.20	-0.04	0.10	0.26	0.30
0	-0.11	0.19	-0.07	0.14	0.02	0.06
0	-0.29	0.32	-0.22	0.25	0.06	0.13
0	-0.18	0.28	-0.17	0.24	0.35	0.38
0	0.08	0.19	0.17	0.22	0.11	0.17
0	0.38	0.42	0.58	0.62	0.07	0.16
0	0.04	0.15	0.03	0.09	-0.02	0.10
0	0.08	0.17	0.15	0.22	-0.01	0.08
0	0.11	0.19	0.16	0.22	0.00	0.03
0	-0.08	0.15	0.00	0.07	0.11	0.18
0	0.44	0.53	0.40	0.48	0.09	0.15
0	-0.06	0.19	-0.02	0.13	-0.22	0.27
0	0.06	0.18	0.00	0.10	0.19	0.22
0	0.16	0.21	0.11	0.17	-0.01	0.08
0	0.19	0.23	0.08	0.14	0.05	0.15
0	0.10	0.21	0.06	0.15	0.37	0.42
0	0.15	0.21	0.16	0.22	-0.06	0.13
0	0.16	0.24	0.18	0.24	0.00	0.07
0	0.44	0.46	0.59	0.62	0.05	0.09
0	0.06	0.24	0.09	0.19	0.16	0.22
0	-0.31	0.36	-0.29	0.33	0.08	0.19
0	-0.08	0.17	-0.01	0.07	0.00	0.02
0	-0.14	0.20	-0.06	0.12	0.18	0.26
0	-0.04	0.19	0.04	0.13	0.11	0.21
0	0.15	0.28	0.12	0.23	-0.43	0.47
0	0.17	0.21	0.30	0.34	0.45	0.59
0	0.16	0.27	0.15	0.24	0.37	0.42
0	0.14	0.19	0.13	0.19	-0.18	0.24
0	-0.09	0.17	-0.05	0.09	0.14	0.24
0	-0.01	0.21	0.09	0.19	-0.49	0.53
0	0.50	0.65	0.52	0.64	0.01	0.08
0	-0.24	0.32	-0.21	0.27	0.34	0.37
0	0.11	0.22	0.08	0.18	0.04	0.13
0	-0.21	0.25	-0.03	0.08	-0.11	0.17

https://doi.org/10.26219/heal.aueb.5951

0	-0.38	0.41	-0.37	0.39	0.29	0.33
0	0.22	0.28	0.30	0.34	0.02	0.10
0	-0.21	0.27	-0.19	0.24	0.22	0.27
0	-0.06	0.13	0.13	0.18	0.14	0.19
0	-0.15	0.19	-0.09	0.13	-0.03	0.09
0	-0.18	0.21	-0.06	0.11	0.56	0.58
0	0.16	0.24	0.13	0.22	0.04	0.09
0	0.02	0.23	0.02	0.13	0.38	0.42
0	0.29	0.38	0.31	0.40	0.50	0.52

Table 4.2.3: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Adding correlation to our data improves the estimations of the coefficients. Especially, better values of the RMSE are observed for all values of λ. The value of λ=4.9, which is the one suggested by Candes and Tao, has the lower RMSE.

Lasso (running time=42 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	33.25	20.48	42.19	29.56
5	4.32	1.19	3.67	1.68
7	6.83	1.01	6.69	1.09
8	3.18	4.91	2.28	5.80
6	7.29	1.60	7.42	1.70
9	9.93	1.23	10.11	1.41
5	0.02	4.98	0.00	5.00
7	0.00	7.00	0.00	7.00
7	6.32	1.01	5.90	1.33
6	3.62	2.48	2.79	3.28
4	2.81	1.49	2.23	2.00
0	1.85	2.10	1.77	2.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.01	0.00	0.00
0	2.28	2.46	2.54	2.71
0	0.17	0.40	0.31	0.59
0	0.00	0.00	0.00	0.00
0	0.30	0.52	0.30	0.53
0	0.00	0.00	0.00	0.00
0	0.01	0.07	0.01	0.04
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.02	0.29	0.27	0.63
0	0.00	0.10	0.00	0.00
0	0.00	0.23	0.05	0.17
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.12	0.00	0.00	0.00
0	0.03	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.01
0	0.04	0.14	0.02	0.09
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.11	0.23	0.01	0.06
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.11	0.27	0.06	0.18
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	2.40	2.68	3.22	3.45
0	0.39	0.61	0.21	0.41
0	0.00	0.00	0.00	0.00
0	1.11	1.46	1.53	1.83
0	0.00	0.00	0.00	0.00
0	0.05	0.14	0.01	0.03
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.22	1.46	0.91	1.19
0	0.00	0.00	0.00	0.00
0	0.00	0.01	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.03	0.12	0.00	0.04
0	0.01	0.07	0.01	0.08
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.62	0.94	1.05	1.31
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.05	1.22	1.25	1.41
0	0.59	0.81	0.52	0.74
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.24	0.41	0.31	0.47
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.32	1.56	1.27	1.51
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.03	1.23	0.87	1.08

Table 4.2.4: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

In this case Lasso does not set to zero the estimations of the non-zero coefficients, which is the good news. Consequently, better values are also observed for the RMSE of those coefficients. On the other hand, we have to mention that it offers non-zero estimates for some of the zero coefficients. That means that Lasso suggests us to think of these predictors as they should be included in the model. This is completely wrong as we know that they do not participate in producing the data.

- n=25 and r=0.96

Dantzig selector

	λ=0.001 (running time=7800 s)		λ=$\sigma\sqrt{2logp}$=4.9 (running time=25800 s)		λ=10 (running time=24900 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	10.26	2.95	8.95	4.17	8.11	4.99
5	4.04	1.12	3.95	1.17	4.00	1.12
7	7.58	0.87	7.49	0.78	7.35	0.72
8	7.03	1.15	7.20	0.97	7.11	1.07
6	6.40	0.71	6.06	0.56	5.93	0.55
9	10.31	1.43	9.86	0.99	9.76	0.91
5	5.19	0.52	5.03	0.40	4.87	0.44

https://doi.org/10.26219/heal.aueb.5951

7	0.75	6.28	0.31	21.17	0.20	6.81
7	5.85	1.37	5.77	1.43	6.06	1.19
6	7.21	1.28	7.42	1.48	7.48	1.54
4	4.02	0.47	3.77	0.47	3.56	0.61
0	0.88	0.89	0.98	0.99	0.96	0.97
0	0.39	0.41	0.36	0.38	0.29	0.33
0	0.56	0.61	0.55	0.62	0.56	0.63
0	-0.08	0.15	-0.05	0.11	-0.03	0.09
0	-0.37	0.40	-0.43	0.47	-0.43	0.46
0	-0.09	0.18	-0.04	0.11	-0.02	0.09
0	-0.06	0.12	-0.03	0.08	-0.01	0.05
0	0.18	0.23	0.15	0.20	0.13	0.18
0	0.01	0.16	0.01	0.10	0.01	0.09
0	0.32	0.36	0.30	0.33	0.32	0.36
0	0.80	0.82	0.77	0.79	0.75	0.77
0	0.02	0.16	0.03	0.11	0.04	0.10
0	0.05	0.15	0.10	0.16	0.10	0.17
0	0.34	0.36	0.42	0.44	0.42	0.45
0	-0.01	0.17	-0.01	0.14	0.00	0.11
0	0.16	0.26	0.11	0.19	0.10	0.19
0	0.32	0.40	0.29	0.37	0.32	0.41
0	0.02	0.18	0.03	0.13	0.05	0.13
0	0.19	0.26	0.23	0.29	0.25	0.30
0	-0.11	0.23	-0.07	0.15	-0.04	0.13
0	0.06	0.17	0.03	0.12	0.02	0.10
0	0.20	0.28	0.20	0.27	0.18	0.24
0	-0.21	0.24	-0.12	0.17	-0.07	0.12
0	0.13	0.21	-0.08	0.15	-0.04	0.10
0	0.29	0.38	0.17	0.27	0.11	0.20
0	0.04	0.14	0.06	0.12	0.03	0.08
0	-0.02	0.18	0.04	0.13	0.05	0.14
0	0.15	0.21	0.12	0.18	0.11	0.17
0	-0.31	0.35	-0.32	0.35	-0.28	0.31
0	-0.22	0.25	-0.14	0.19	-0.12	0.16
0	0.27	0.31	0.31	0.35	0.32	0.38
0	0.34	0.38	0.40	0.43	0.41	0.44
0	-0.16	0.28	-0.10	0.20	-0.05	0.16
0	-0.07	0.20	-0.05	0.14	-0.03	0.12
0	0.05	0.12	0.03	0.07	0.01	0.05
0	0.10	0.20	0.06	0.16	0.05	0.15
0	-0.25	0.33	-0.21	0.28	-0.18	0.24
0	-0.08	0.18	-0.06	0.14	-0.06	0.13
0	0.18	0.26	0.12	0.21	0.10	0.19
0	-0.10	0.17	-0.02	0.09	-0.01	0.07
0	-0.17	0.21	-0.07	0.12	-0.03	0.09
0	-0.14	0.19	-0.12	0.17	-0.10	0.17
0	0.11	0.37	0.12	0.31	0.14	0.30
0	-0.01	0.24	0.01	0.16	0.03	0.14
0	0.09	0.22	0.04	0.13	0.03	0.12
0	-0.05	0.16	0.00	0.08	0.02	0.10

https://doi.org/10.26219/heal.aueb.5951

0	0.16	0.20	0.06	0.11	0.03	0.07
0	0.16	0.21	0.13	0.18	0.11	0.17
0	0.35	0.39	0.46	0.48	0.48	0.51
0	-0.06	0.15	-0.01	0.08	0.02	0.09
0	0.00	0.14	0.06	0.14	0.06	0.12
0	0.01	0.16	0.02	0.12	0.02	0.10
0	0.10	0.16	0.07	0.13	0.05	0.11
0	0.06	0.17	0.04	0.11	0.03	0.10
0	0.12	0.19	0.05	0.13	0.03	0.10
0	-0.05	0.16	-0.13	0.19	-0.15	0.20
0	0.24	0.27	0.29	0.32	0.29	0.32
0	0.02	0.13	-0.01	0.11	-0.02	0.10
0	-0.15	0.21	-0.08	0.14	-0.04	0.12
0	0.12	0.22	0.11	0.20	0.10	0.18
0	-0.04	0.16	0.02	0.14	0.04	0.14
0	0.35	0.38	0.37	0.39	0.37	0.40
0	0.30	0.36	0.29	0.36	0.31	0.39
0	-0.01	0.14	0.04	0.10	0.04	0.09
0	0.13	0.18	0.14	0.20	0.12	0.17
0	-0.18	0.23	-0.06	0.14	0.00	0.08
0	0.44	0.47	0.41	0.45	0.35	0.39
0	-0.21	0.25	-0.16	0.21	-0.11	0.17
0	-0.12	0.18	-0.15	0.20	-0.14	0.19
0	-0.10	0.17	-0.06	0.13	-0.05	0.12
0	-0.21	0.25	-0.15	0.18	-0.11	0.15
0	0.17	0.24	0.20	0.27	0.21	0.27
0	-0.01	0.17	0.00	0.11	0.01	0.09
0	0.52	0.55	0.66	0.70	0.72	0.75
0	0.01	0.12	0.01	0.06	0.01	0.06
0	-0.06	0.17	0.00	0.11	0.01	0.10
0	0.02	0.12	0.01	0.07	0.01	0.06
0	-0.14	0.20	-0.08	0.14	-0.04	0.08
0	0.34	0.37	0.37	0.42	0.40	0.42
0	0.28	0.34	0.16	0.22	0.10	0.16
0	-0.09	0.21	-0.06	0.15	-0.06	0.13
0	-0.18	0.24	-0.11	0.16	-0.08	0.13
0	0.18	0.27	0.16	0.25	0.14	0.22
0	-0.20	0.24	-0.13	0.18	-0.10	0.15
0	-0.10	0.18	-0.03	0.11	-0.01	0.08
0	-0.16	0.18	-0.05	0.09	-0.02	0.06
0	0.00	0.19	0.05	0.15	0.06	0.17
0	-0.08	0.18	-0.01	0.14	-0.01	0.09
0	0.11	0.16	0.11	0.15	0.09	0.13
0	0.10	0.21	0.15	0.22	0.18	0.25

Table 4.2.5: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Putting correlation close to one improves even more the results of Dantzig selector. Better estimations and smaller values of the RMSEs are observed for all

https://doi.org/10.26219/heal.aueb.5951

values of λ. The λ that is suggested by Candes and Tao (here λ=4.9) gives the best results.

Lasso (running time=45 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	21.53	8.62	30.70	17.80
5	0.00	5.00	0.00	5.00
7	0.01	6.99	0.00	7.00
8	6.71	1.48	6.38	1.77
6	0.00	6.00	0.00	6.00
9	1.46	7.57	1.22	7.81
5	1.32	3.71	0.74	4.29
7	0.00	7.00	0.00	7.00
7	3.63	3.50	3.74	3.39
6	4.12	1.96	3.65	2.41
4	1.43	2.63	0.84	3.19
0	0.30	0.42	0.07	0.17
0	0.00	0.00	0.00	0.00
0	1.03	1.22	1.04	1.23
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.09	0.03	0.12
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	4.37	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.01	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	4.44	4.37	4.43
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	3.87	3.99	4.15	4.24
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.01	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	7.78	2.47	8.26	8.29
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.02	0.00	0.01
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	2.12	2.26	2.14	2.27
0	0.00	0.07	0.01	0.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	5.13	5.18	5.77	5.82
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.01	0.04	0.01	0.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	8.74	8.77	8.90	8.92
0	0.00	0.00	0.00	0.00
0	1.68	1.87	1.71	1.86
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.01	0.05	0.00	0.03
0	0.00	0.00	0.00	0.00
0	0.59	0.78	0.36	0.55
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.50	1.73	1.38	1.63
0	0.00	0.00	0.00	0.00
0	2.23	2.29	2.00	2.07
0	0.00	0.00	0.00	0.00
0	1.65	1.81	1.70	1.86
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.65	1.83	1.46	1.66
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.02	0.00	0.00
0	0.00	0.00	0.00	0.00

Table 4.2.6: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=25 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Estimations of Lasso are not good enough for the case of r=0.96. Although the majority of the estimations is set to zero with zero RMSE, we have many non-zero estimations for zero coefficients. Consequently, large values of the RMSE are observed for coefficients that should not be included in the model.

Summing the above results that refer to the case when n=25, we could say that adding correlation between the variables improves the results arising from Dantzig selector. Values of the RMSE are getting lower, especially for λ=σ$\sqrt{2logp}$ =4.9. This improvement is not observed for the method of Lasso.

Now we try to make the problem more difficult by using less number of observations. We use 10 observations and we study the results arising for the different values of r.

- n=10 and r=0

Dantzig selector

	λ=0.001 (running time=3900s)		λ=σ$\sqrt{2logp}$=4.9 (running time=18600 s)		λ=10 (running time=16680 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	0.05	12.96	0.00	13.00	0.00	13.00
5	6.48	2.27	6.24	2.03	6.24	2.07
7	5.13	2.62	4.90	2.73	4.83	2.77
8	5.08	3.81	4.93	4.03	4.93	3.96
6	5.95	2.24	5.90	2.32	5.78	2.24
9	8.20	1.59	8.21	1.44	8.13	1.52
5	5.94	1.62	5.93	1.62	5.97	1.65
7	0.64	6.47	0.67	6.38	0.76	6.31
7	9.30	2.90	9.30	3.01	9.16	2.83
6	6.92	1.84	6.73	1.67	6.88	1.75
4	3.42	1.28	3.52	1.26	3.42	1.27
0	0.11	0.16	0.12	0.16	0.12	0.15

https://doi.org/10.26219/heal.aueb.5951

0	-0.01	0.11	0.02	0.13	0.03	0.12
0	0.07	0.13	0.07	0.13	0.06	0.10
0	0.14	0.19	0.17	0.21	0.15	0.20
0	0.08	0.14	0.09	0.15	0.09	0.17
0	0.02	0.16	0.04	0.16	0.04	0.14
0	0.01	0.09	0.03	0.08	0.03	0.07
0	0.06	0.11	0.07	0.11	0.06	0.11
0	-0.01	0.16	0.01	0.13	0.02	0.11
0	0.12	0.15	0.13	0.16	0.13	0.15
0	0.21	0.31	0.22	0.33	0.21	0.31
0	0.14	0.31	0.13	0.27	0.14	0.28
0	0.28	0.44	0.26	0.42	0.28	0.43
0	0.06	0.17	0.05	0.13	0.07	0.15
0	0.28	0.36	0.28	0.37	0.29	0.37
0	0.00	0.10	0.02	0.09	0.03	0.09
0	0.16	0.27	0.15	0.24	0.17	0.26
0	0.18	0.29	0.22	0.35	0.19	0.30
0	0.26	0.32	0.28	0.35	0.25	0.31
0	0.02	0.10	0.03	0.08	0.04	0.09
0	0.03	0.11	0.04	0.12	0.05	0.11
0	0.28	0.37	0.28	0.36	0.27	0.37
0	0.26	0.43	0.24	0.42	0.21	0.37
0	0.01	0.10	0.02	0.07	0.03	0.08
0	0.12	0.22	0.14	0.25	0.15	0.26
0	0.06	0.13	0.08	0.13	0.07	0.12
0	-0.03	0.16	-0.01	0.11	0.00	0.11
0	0.24	0.36	0.23	0.35	0.26	0.38
0	0.13	0.21	0.15	0.22	0.13	0.19
0	0.23	0.35	0.22	0.36	0.22	0.33
0	0.16	0.20	0.16	0.20	0.17	0.21
0	-0.07	0.12	-0.05	0.10	-0.04	0.09
0	0.02	0.11	0.03	0.10	0.03	0.09
0	0.08	0.18	0.09	0.18	0.09	0.18
0	0.13	0.19	0.13	0.18	0.14	0.20
0	0.01	0.15	0.02	0.12	0.03	0.10
0	0.11	0.15	0.11	0.15	0.11	0.15
0	0.07	0.19	0.08	0.21	0.10	0.23
0	0.13	0.21	0.15	0.22	0.15	0.25
0	0.16	0.20	0.16	0.19	0.17	0.21
0	0.06	0.12	0.07	0.13	0.08	0.14
0	0.09	0.15	0.07	0.12	0.07	0.10
0	0.11	0.23	0.12	0.23	0.14	0.28
0	0.07	0.15	0.07	0.13	0.07	0.14
0	0.03	0.16	0.07	0.17	0.07	0.17
0	0.07	0.15	0.09	0.15	0.09	0.15
0	0.02	0.11	0.04	0.08	0.04	0.09
0	0.00	0.11	0.01	0.09	0.02	0.09
0	0.27	0.38	0.27	0.38	0.28	0.39
0	0.06	0.14	0.07	0.13	0.06	0.10
0	0.08	0.24	0.12	0.25	0.11	0.24

https://doi.org/10.26219/heal.aueb.5951

0	0.09	0.13	0.10	0.13	0.10	0.15
0	0.09	0.13	0.10	0.15	0.10	0.14
0	0.24	0.38	0.21	0.35	0.24	0.39
0	0.26	0.34	0.28	0.36	0.29	0.38
0	0.09	0.14	0.08	0.13	0.10	0.16
0	0.17	0.24	0.17	0.24	0.17	0.24
0	0.15	0.26	0.16	0.28	0.15	0.25
0	0.06	0.25	0.07	0.23	0.06	0.20
0	0.10	0.18	0.11	0.18	0.12	0.20
0	0.16	0.19	0.17	0.20	0.16	0.19
0	0.01	0.08	0.02	0.08	0.02	0.07
0	0.10	0.16	0.12	0.18	0.12	0.19
0	0.08	0.13	0.10	0.14	0.08	0.11
0	0.14	0.20	0.15	0.21	0.15	0.20
0	0.03	0.11	0.04	0.09	0.05	0.13
0	0.04	0.17	0.02	0.09	0.03	0.11
0	0.07	0.16	0.11	0.21	0.10	0.19
0	0.12	0.23	0.11	0.20	0.12	0.22
0	0.06	0.18	0.08	0.19	0.08	0.20
0	-0.03	0.11	-0.01	0.08	-0.01	0.07
0	0.03	0.11	0.05	0.12	0.05	0.10
0	0.36	0.50	0.34	0.47	0.34	0.47
0	0.13	0.28	0.13	0.25	0.12	0.24
0	0.05	0.12	0.05	0.13	0.05	0.12
0	0.14	0.22	0.15	0.25	0.14	0.21
0	0.04	0.09	0.05	0.10	0.05	0.09
0	0.17	0.21	0.16	0.20	0.16	0.20
0	0.19	0.31	0.20	0.30	0.21	0.32
0	0.06	0.11	0.06	0.10	0.07	0.12
0	0.18	0.24	0.18	0.22	0.20	0.25
0	-0.01	0.11	0.01	0.09	0.02	0.10
0	0.14	0.19	0.14	0.17	0.13	0.17
0	0.10	0.20	0.10	0.18	0.10	0.20
0	-0.13	0.17	-0.10	0.16	-0.08	0.13
0	-0.05	0.11	0.00	0.10	0.00	0.08
0	0.16	0.19	0.17	0.21	0.16	0.19
0	0.09	0.13	0.10	0.14	0.09	0.12
0	0.16	0.22	0.17	0.22	0.15	0.21
0	0.10	0.17	0.10	0.17	0.11	0.18

Table 4.2.7: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

The constant cannot be estimated effectively by Dantzig selector in this case. However, Dantzig selector offers estimates close to zero for zero coefficients and it also gives good values for the corresponding RMSEs. Only a few mistakes are observed at the estimations of non-zero coefficients.

https://doi.org/10.26219/heal.aueb.5951

Lasso (running time=38 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	300.69	907.16	303.71	291.18
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
8	0.00	8.00	0.00	8.00
6	0.00	6.00	0.00	6.00
9	0.00	9.00	0.00	9.00
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
7	0.00	7.00	0.00	7.00
6	0.00	6.00	0.00	6.00
4	0.00	4.00	0.00	4.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.04	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.01	0.00	0.00
0	0.01	0.06	0.00	0.00
0	0.02	0.14	0.00	0.00
0	0.22	0.37	0.00	0.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-0.06	0.24	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-4.75	0.15	-1.83	7.69
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.04	0.20	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-0.03	0.16	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	-2.03	2.23	-0.42	0.99
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.02	0.08	0.00	0.00
0	-0.19	0.41	-0.01	0.05
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.02	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.50	0.81	0.01	0.32
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.79	1.12	0.22	0.51
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	7.02	7.21	3.98	4.99
0	-4.11	4.16	-1.48	2.15
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	5.58	5.79	2.97	3.78
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.05	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0,00	0.00	0.00

Table 4.2.8: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso seems to have great difficulty in estimating coefficients when the number of observations is small relative to the number of predictors (n<<p). It sets to zero the estimations of almost all the coefficients, and the non-zero estimations do not correspond to the correct ones. This might happen due to the collinearity which might exist between the variables. As a result of the above, we consider some variables as important even if they should not be included in the model.

- n=10 and r=0.5

Dantzig selector

	λ=0.001 (running time=4620 s)		λ=$\sigma\sqrt{2logp}$=4.9 (running time=20520 s)		λ=10 (running time=18060 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	1.13	11.94	0.00	13.00	0.00	13.00
5	5.18	1.22	5.01	1.23	4.92	1.19
7	7.94	1.89	7.86	1.81	7.81	1.85
8	6.54	3.10	6.87	2.94	7.06	3.00
6	6.48	1.47	6.44	1.50	6.45	1.56
9	8.68	1.74	8.75	1.73	8.80	1.89
5	4.07	2.00	3.91	2.02	3.90	2.01
7	1.57	5.53	1.37	5.72	1.34	5.74
7	7.05	1.44	7.18	1.43	7.13	1.37
6	8.61	3.71	7.97	3.28	7.71	3.13
4	5.75	2.35	5.61	2.22	5.53	2.15
0	0.14	0.22	0.14	0.20	0.12	0.17
0	0.11	0.20	0.11	0.16	0.14	0.21
0	0.06	0.15	0.07	0.12	0.08	0.13
0	0.07	0.15	0.08	0.14	0.08	0.14
0	-0.01	0.10	0.01	0.07	0.02	0.11
0	0.04	0.18	0.05	0.17	0.03	0.12
0	0.07	0.12	0.08	0.13	0.07	0.13
0	0.08	0.19	0.08	0.16	0.07	0.15
0	0.13	0.20	0.13	0.19	0.12	0.18
0	-0.08	0.12	-0.03	0.09	-0.01	0.09
0	0.13	0.18	0.11	0.18	0.11	0.17
0	-0.03	0.19	0.00	0.15	0.02	0.15
0	-0.04	0.19	-0.01	0.15	0.00	0.12

https://doi.org/10.26219/heal.aueb.5951

0	0.04	0.13	0.04	0.11	0.04	0.12
0	0.06	0.10	0.05	0.11	0.06	0.12
0	-0.01	0.19	0.00	0.15	0.00	0.14
0	0.19	0.23	0.20	0.24	0.21	0.24
0	0.03	0.11	0.05	0.10	0.07	0.13
0	-0.13	0.19	-0.08	0.14	-0.01	0.12
0	0.17	0.21	0.19	0.23	0.18	0.22
0	0.10	0.18	0.11	0.15	0.11	0.16
0	0.03	0.12	0.03	0.09	0.04	0.10
0	0.10	0.24	0.10	0.22	0.08	0.21
0	-0.03	0.15	-0.01	0.10	0.00	0.10
0	0.19	0.30	0.20	0.32	0.19	0.30
0	-0.05	0.14	-0.04	0.11	-0.04	0.13
0	-0.01	0.14	0.03	0.12	0.04	0.11
0	0.05	0.15	0.05	0.12	0.05	0.12
0	-0.05	0.15	-0.02	0.13	0.00	0.11
0	0.08	0.15	0.08	0.14	0.06	0.11
0	0.28	0.31	0.32	0.35	0.31	0.33
0	-0.05	0.11	-0.01	0.10	0.01	0.11
0	0.14	0.20	0.14	0.20	0.12	0.17
0	-0.04	0.10	-0.01	0.06	0.00	0.06
0	-0.10	0.17	-0.07	0.16	-0.01	0.13
0	0.12	0.19	0.10	0.17	0.10	0.16
0	0.06	0.11	0.06	0.11	0.07	0.12
0	-0.09	0.20	-0.05	0.17	-0.02	0.15
0	0.01	0.16	0.04	0.14	0.04	0.15
0	0.12	0.19	0.12	0.17	0.09	0.15
0	0.08	0.13	0.06	0.11	0.06	0.10
0	0.28	0.37	0.29	0.35	0.28	0.34
0	0.03	0.10	0.04	0.12	0.04	0.12
0	-0.01	0.12	0.02	0.11	0.02	0.12
0	0.03	0.10	0.04	0.09	0.03	0.08
0	-0.25	0.28	-0.24	0.29	-0.21	0.26
0	-0.02	0.14	-0.01	0.10	-0.01	0.09
0	0.11	0.15	0.09	0.12	0.09	0.15
0	0.08	0.15	0.08	0.14	0.07	0.13
0	0.10	0.14	0.11	0.16	0.11	0.15
0	0.04	0.17	0.04	0.11	0.05	0.12
0	0.18	0.21	0.18	0.21	0.17	0.22
0	0.10	0.16	0.09	0.17	0.08	0.15
0	-0.06	0.19	-0.03	0.13	0.00	0.12
0	0.09	0.17	0.10	0.16	0.11	0.15
0	-0.02	0.15	0.02	0.11	0.02	0.10
0	-0.02	0.12	0.00	0.08	0.01	0.09
0	0.06	0.14	0.08	0.15	0.08	0.16
0	0.00	0.12	0.03	0.14	0.03	0.11
0	0.09	0.15	0.10	0.15	0.08	0.14
0	0.20	0.26	0.22	0.27	0.21	0.25
0	0.10	0.21	0.11	0.20	0.10	0.21
0	0.07	0.14	0.08	0.13	0.08	0.14

https://doi.org/10.26219/heal.aueb.5951

0	0.18	0.24	0.20	0.25	0.19	0.23
0	0.04	0.10	0.05	0.10	0.07	0.13
0	-0.06	0.15	-0.01	0.13	0.00	0.11
0	0.10	0.15	0.10	0.14	0.11	0.16
0	-0.02	0.16	-0.01	0.12	0.01	0.11
0	0.07	0.14	0.07	0.12	0.06	0.15
0	0.12	0.17	0.11	0.15	0.11	0.15
0	-0.10	0.16	-0.06	0.11	-0.02	0.11
0	0.07	0.20	0.07	0.15	0.07	0.14
0	0.24	0.29	0.26	0.31	0.24	0.28
0	0.06	0.14	0.08	0.14	0.10	0.17
0	0.01	0.12	0.05	0.11	0.06	0.13
0	0.09	0.20	0.11	0.18	0.11	0.19
0	0.03	0.12	0.06	0.11	0.06	0.11
0	-0.07	0.21	-0.02	0.18	-0.01	0.16
0	-0.05	0.14	-0.02	0.07	0.00	0.09
0	-0.04	0.15	-0.02	0.11	-0.01	0.12
0	-0.03	0.15	-0.01	0.12	0.01	0.13
0	0.13	0.21	0.13	0.18	0.12	0.18
0	0.08	0.19	0.09	0.20	0.07	0.14
0	0.03	0.21	0.02	0.14	0.03	0.15
0	-0.01	0.14	0.01	0.12	0.02	0.13
0	0.17	0.37	0.15	0.33	0.14	0.32
0	0.17	0.24	0.16	0.27	0.14	0.25
0	0.03	0.11	0.06	0.12	0.06	0.13
0	-0.02	0.09	0.09	0.09	0.01	0.06
0	0.06	0.14	0.09	0.15	0.11	0.16

Table 4.2.9: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Adding correlation between the variables seems to improve the results of Dantzig selector, as it gives better estimations with smaller RMSEs, especially for the case of $\lambda=\sigma\sqrt{2logp}$=4.9.

Lasso (running time=33 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	88.90	76.38	125.82	113.23
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
8	11.86	4.00	10.70	2.90
6	0.00	6.00	0.00	6.00
9	1.02	8.04	0.01	8.99
5	0.03	4.98	0.00	5.00
7	0.00	7.00	0.00	7.00
7	0.00	7.00	0.00	7.00

https://doi.org/10.26219/heal.aueb.5951

6	0.00	6.00	0.00	6.00
4	0.00	4.00	0.00	4.00
0	0.00	0.03	0.02	0.20
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.04	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	11.12	11.21	10.02	10.09
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	16.77	16.80	16.48	16.50
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.44	0.87	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.12	0.36	0.00	0.04
0	0.05	0.33	0.22	0.60
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

https://doi.org/10.26219/heal.aueb.5951

0	0.94	1.36	0.01	0.10
0	5.34	5.53	4.24	4.40
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.03	0.17	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.30	1.62	0.02	0.13
0	0.09	0.25	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.23	0.64	0.05	0.24
0	0.00	0.00	0.00	0.00
0	0.09	0.27	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

Table 4.2.10: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0.5 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso does not make an improvement. Although the RMSEs for the most of the coefficients are zero, it continues to suggest wrong variables for keeping in mind for the model.

https://doi.org/10.26219/heal.aueb.5951

- n=10 and r=0.96

Dantzig selector

	λ=0.001 (running time=3960 s)		$\lambda=\sigma\sqrt{2logp}$=4.9 (running time=13800 s)		λ=10 (running time=10800 s)	
True coefficients	Estimates	RMSE	Estimates	RMSE	Estimates	RMSE
13	9.95	3.46	0.00	13.00	0.00	13.00
5	2.18	3.06	1.56	3.61	1.59	3.58
7	6.43	0.90	8.19	1.84	8.43	2.06
8	7.87	1.11	5.42	2.97	5.34	3.01
6	6.95	1.19	8.20	2.43	8.09	2.31
9	6.51	2.66	8.69	0.77	8.61	0.81
5	6.23	1.53	0.57	4.49	0.64	4.42
7	4.55	2.69	2.74	4.47	3.07	4.16
7	4.78	2.75	8.32	1.83	8.56	1.97
6	3.93	2.29	4.33	1.94	4.32	1.97
4	5.41	1.53	4.12	0.87	4.11	0.86
0	0.20	0.26	-0.24	0.44	-0.18	0.39
0	0.12	0.19	0.16	0.19	0.14	0.17
0	0.08	0.33	0.36	0.41	0.34	0.40
0	0.20	0.32	0.05	0.18	0.05	0.17
0	0.05	0.19	0.53	0.62	0.45	0.56
0	0.19	0.28	0.44	0.50	0.41	0.46
0	0.03	0.12	0.16	0.21	0.14	0.19
0	0.15	0.28	0.19	0.35	0.17	0.34
0	0.11	0.15	0.25	0.32	0.23	0.31
0	0.24	0.30	0.02	0.10	0.02	0.08
0	0.06	0.12	0.06	0.12	0.06	0.13
0	-0.01	0.11	0.19	0.28	0.16	0.24
0	0.13	0.20	0.38	0.41	0.36	0.40
0	0.02	0.11	0.01	0.08	0.02	0.09
0	0.03	0.12	0.11	0.15	0.10	0.15
0	0.11	0.22	0.20	0.22	0.18	0.21
0	-0.12	0.15	0.02	0.09	0.01	0.08
0	0.15	0.21	0.40	0.47	0.39	0.47
0	0.14	0.20	0.15	0.21	0.14	0.18
0	0.11	0.19	0.12	0.25	0.11	0.25
0	0.10	0.13	0.01	0.05	0.00	0.02
0	-0.01	0.13	0.05	0.09	0.05	0.09
0	0.02	0.14	0.00	0.01	0.00	0.00
0	0.04	0.14	0.02	0.08	0.01	0.04
0	0.20	0.29	0.27	0.37	0.25	0.34
0	0.82	0.91	0.03	0.15	0.03	0.11
0	0.06	0.17	0.15	0.32	0.13	0.30
0	0.11	0.38	0.04	0.11	0.04	0.11
0	-0.08	0.16	0.12	0.16	0.11	0.15
0	0.19	0.34	0.07	0.13	0.07	0.15
0	0.02	0.11	1.10	1.10	1.08	1.09

https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.08	0.03	0.09	0.03	0.11
0	0.06	0.20	0.11	0.15	0.09	0.11
0	0.21	0.28	0.23	0.25	0.21	0.24
0	0.09	0.15	0.00	0.01	0.00	0.01
0	0.09	0.18	0.32	0.47	0.30	0.46
0	-0.03	0.12	0.28	0.30	0.27	0.29
0	0.07	0.12	0.17	0.25	0.15	0.21
0	0.04	0.23	0.10	0.14	0.08	0.11
0	0.35	0.40	0.04	0.13	0.04	0.11
0	0.00	0.12	0.01	0.04	0.01	0.03
0	0.07	0.20	0.19	0.21	0.17	0.19
0	0.18	0.30	0.31	0.35	0.28	0.33
0	0.06	0.17	0.12	0.17	0.10	0.15
0	0.11	0.15	0.21	0.26	0.20	0.26
0	-0.01	0.09	0.04	0.07	0.03	0.06
0	0.11	0.15	0.28	0.37	0.24	0.33
0	0.11	0.16	0.12	0.21	0.12	0.19
0	0.05	0.12	0.01	0.04	0.02	0.05
0	0.28	0.37	0.17	0.35	0.19	0.37
0	0.10	0.12	0.05	0.08	0.05	0.08
0	0.11	0.17	0.08	0.15	0.08	0.15
0	0.14	0.19	0.12	0.14	0.11	0.13
0	-0.15	0.18	0.10	0.16	0.10	0.16
0	0.13	0.19	0.13	0.19	0.11	0.17
0	-0.07	0.12	0.17	0.27	0.16	0.26
0	0.22	0.24	0.03	0.06	0.03	0.07
0	0.06	0.11	0.02	0.09	0.03	0.13
0	0.07	0.14	0.07	0.11	0.07	0.11
0	-0.06	0.13	0.04	0.08	0.03	0.07
0	-0.01	0.08	0.18	0.30	0.20	0.33
0	0.11	0.16	0.15	0.20	0.14	0.17
0	-0.16	0.19	0.11	0.17	0.12	0.19
0	0.26	0.30	0.23	0.27	0.23	0.26
0	0.13	0.26	-0.02	0.09	-0.01	0.09
0	0.24	0.28	0.01	0.07	0.01	0.05
0	0.23	0.32	0.05	0.11	0.06	0.14
0	0.12	0.15	0.22	0.30	0.20	0.27
0	0.09	0.18	0.15	0.20	0.14	0.18
0	0.05	0.13	0.01	0.06	0.02	0.06
0	0.05	0.11	0.15	0.20	0.14	0.21
0	0.08	0.13	0.09	0.17	0.09	0.18
0	0.15	0.21	0.11	0.21	0.12	0.24
0	0.28	0.32	0.38	0.43	0.35	0.39
0	0.32	0.49	0.02	0.04	0.02	0.05
0	0.01	0.13	0.08	0.12	0.08	0.12
0	0.04	0.09	0.04	0.15	0.04	0.15
0	0.03	0.14	0.23	0.26	0.21	0.24
0	0.03	0.14	-0.01	0.04	0.00	0.03
0	0.20	0.27	0.03	0.07	0.04	0.08
0	-0.13	0.17	0.13	0.17	0.11	0.14

https://doi.org/10.26219/heal.aueb.5951

0	0.03	0.14	0.15	0.26	0.15	0.27
0	-0.03	0.10	0.05	0.12	0.05	0.13
0	0.30	0.33	0.13	0.17	0.12	0.15
0	0.13	0.22	0.07	0.13	0.06	0.13
0	0.06	0.11	0.01	0.03	0.00	0.02
0	-0.06	0.13	0.27	0.39	0.25	0.37
0	0.17	0.25	0.05	0.13	0.05	0.14
0	0.05	0.15	0.13	0.27	0.12	0.25
0	0.21	0.24	0.21	0.31	0.21	0.31

Table 4.2.11: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Putting correlation close to one causes great improvement to the results derived from Dantzig selector. The case with better values of the RMSE is the case of λ=0.001.

Lasso (running time=36 s)

	λ min		λ 1se	
True coefficients	Estimates	RMSE	Estimates	RMSE
13	90.02	77.29	142.19	129.47
5	0.00	5.00	0.00	5.00
7	0.00	7.00	0.00	7.00
8	0.07	7.93	0.01	7.99
6	0.00	6.00	0.00	6.00
9	0.00	9.00	0.00	9.00
5	9.85	4.96	8.78	3.92
7	0.00	7.00	0.00	7.00
7	0.00	7.00	0.00	7.00
6	0.00	1.90	0.00	6.00
4	0.01	3.99	0.00	4.00
0	2.01	2.37	1.87	2.16
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.02	0.08	0.00	0.02
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	5.73	5.87	4.75	0.05
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

0	0.22	0.40	0.05	0.12
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	22.84	22.89	21.81	21.85
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.12	0.43	0.23	0.59
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	6.02	6.35	6.15	6.46
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.44	0.88	0.44	0.87
0	0.00	0.00	0.00	0.00
0	0.09	0.44	0.09	0.43
0	0.00	0.01	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.07	0.17	0.00	0.01
0	0.00	0.01	0.00	0.02
0	0.00	0.00	0.00	0.00
0	0.01	0.03	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	2.41	2.61	1.30	1.53
0	0.00	0.00	0.00	0.00

 https://doi.org/10.26219/heal.aueb.5951

0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	1.07	1.38	0.53	0.88
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	6.58	6.84	7.17	7.41
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.03	0.05	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00

Table 4.2.12: Table with the expected values of the estimates and the RMSE. It refers to the case in which n=10 and r=0.96 and it contains results for 3 different values of λ. True coefficients are also presented for easy comparison.

Lasso is not able to detect which variables should be included in the model as it gives wrong estimates for all of the variables. However, we have to mention that the RMSE is close to zero for all the coefficients that are estimated to be zero, and that is really good.

In general, when n=25 or n=10 and p=100, Dantzig selector seems to give better estimations than Lasso. However, we have to mention that this happen in the case we know the starting values of the coefficients. Adding correlation between the variables improves the estimations and the values of the RMSEs. We could also say that the RMSE of the estimations derived from Lasso is zero for the zero estimates, but it is not useful as Lasso cannot detect which variables should be included in the model.

If we would like to comment on the running time of the algorithms, we could say that Lasso is much faster than Dantzig selector. Factors that affect the running time are the correlation between the variables (r), the value of λ and the number of observations (n). The most remarkable fact for the running time is that $\lambda=\sigma\sqrt{2logp}$

https://doi.org/10.26219/heal.aueb.5951

=4.9 needs more time to give results than the other values of λ. This λ is the one suggested by Candes and Tao and usually gives the best value of the RMSE.

So far, we studied the cases when 8 out of 10 predictors are zero and when 90 out of 100 predictors are zero. Comparing these two cases, we could say that in the first case Lasso gives better RMSEs than Dantzig selector. In the second one, Dantzig selector has again larger values of the RMSE, but it seems that it is able to detect the predictors that should be included in the model. It offers estimates close to zero for the zero coefficients. As we mentioned above, all these comments refer to the case we know the starting values of the coefficients. If we do not know the starting values, the results of Dantzig selector are not good enough.

Before studying the real-world problem, we have to mention that we tried another one simulation scheme in which we use 1000 explanatory variables. 980 out of these 1000 variables have zero coefficients and we expect from Dantzig selector and Lasso to detect which variables should be included in the model. We do not present results of the aforementioned simulation scheme because of the running time of the algorithm. Specifically, we should say that the algorithm needs two and a half days to give results for only one repetition of the above simulation scheme, so it is not easy to study this case.

https://doi.org/10.26219/heal.aueb.5951

Chapter 5: Real-world problem

In this section of our thesis, we are going to deal with a real-world problem in order to see if Dantzig selector operates equally well in this problem as in the simulated problems presented in the previous section. Firstly, we describe the problem and consequently we apply different methods to our data and compare the results.

5.1 Description of the problem

Specifically, we use a dataset called diabetes[11], which contains 9 variables and 768 observations . We have the dependent variable "Class", which shows if a woman is diabetic or no (1 or 0). The remaining variables are the predictors and we try to estimate the coefficients of them by using Dantzig selector. In order to decide if the estimation is good, we also use Lasso and ols estimators and compare the results.

Starting our analysis, we present the predictors[12]:

Pregnant: number of times pregnant

Glucose: plasma glucose concentration a 2 hours in an oral glucose tolerance test

Blood press: diastolic blood pressure (mm Hg)

Triceps: triceps skin fold thickness (mm)

Insulin: 2-hour serum insulin (mu U/ml)

BMI: body mass index ((weight in kg)/(height in m)^2)

Pedigree: diabetes pedigree function

Age: years

Although we are not interested in descriptive analysis, we can take a look of our data by computing the correlation between the variables. In this way, we will be able to know which variables affect our dependent variable and to what extent. Hence, we are waiting for estimations close to zero for the coefficients of variables that do not seem to affect the dependent one. We cannot use a simple correlation matrix because our dependent variable is qualitative. Hence, we study the

[11] The code used is shown in the appendix.

[12] We mention the measurement units where appropriate.

https://doi.org/10.26219/heal.aueb.5951

correlation between the dependent variable and each one of the predictors by using the appropriate boxplots. These boxplots are presented below:

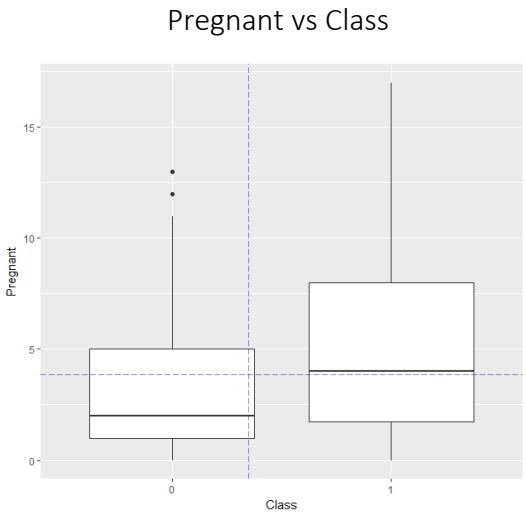

Figure 5.1.1: Boxplots of Pregnant per Class. The median seems to differ between the two classes. That means that Pregnancy affects to which class a woman owns.

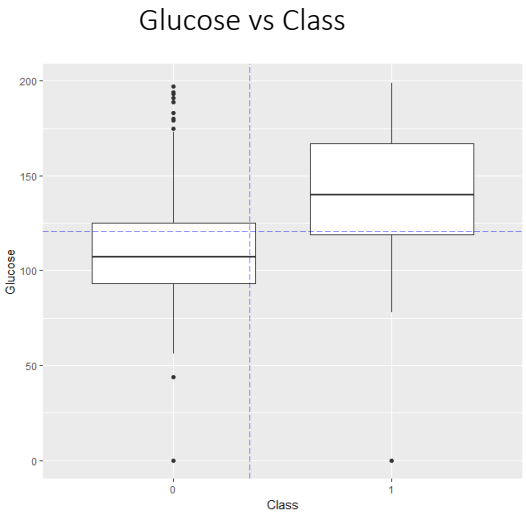

Figure 5.1.2: Boxplots of Glucose per Class. The medians of the classes seem to differ. That means that concentration of Glucose affects to which class a woman owns.

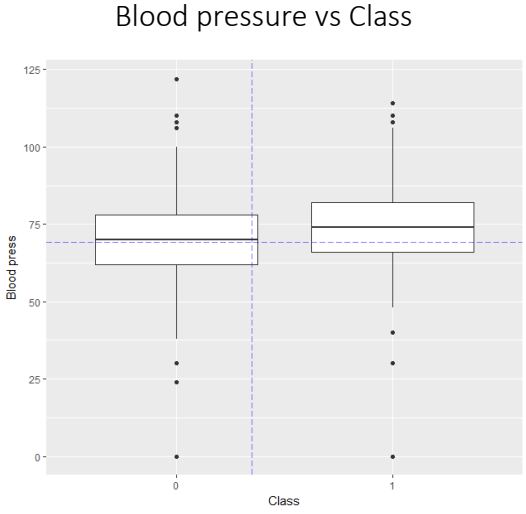

Figure 5.1.3: Boxplots of Blood pressure per Class. The medians do not differ significantly, so blood pressure does not play an important role to our dependent Variable.

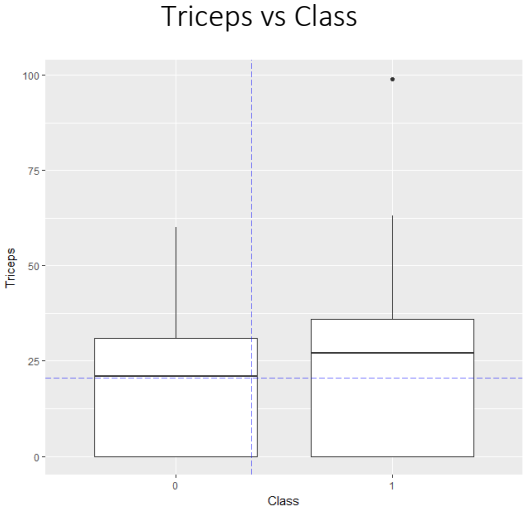

Figure 5.1.4: Boxplots of Triceps per Class. The medians do not differ significantly, so triceps do not play an important role in categorizing diabetics.

https://doi.org/10.26219/heal.aueb.5951

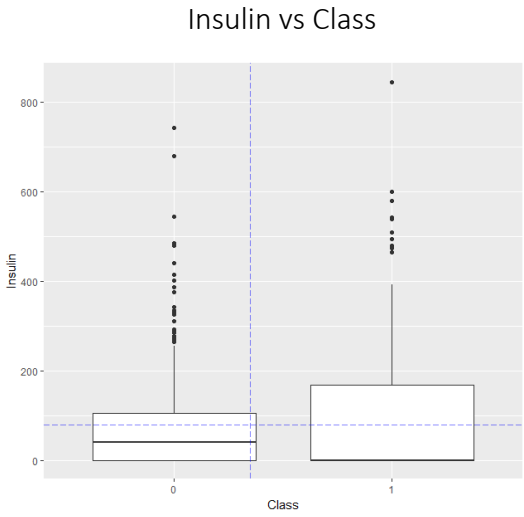

Figure 5.1.5: Boxplots of Insulin per Class. The medians do not differ, so Insulin does not play an important role in categorizing diabetic women.

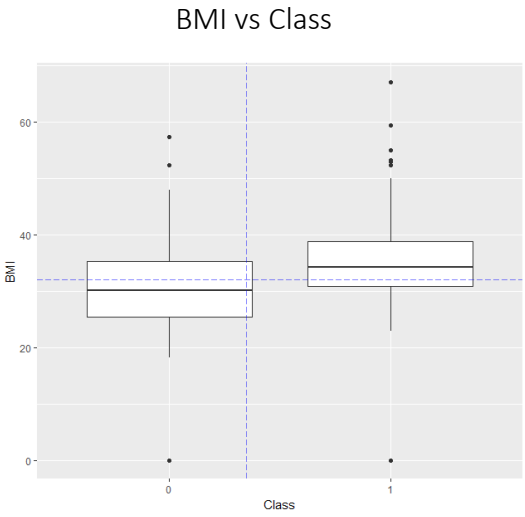

Figure 5.1.6: Boxplots of BMI per Class. The medians differ a little as the median of class 0 is below the Expected value of BMI, while the median of class 1 is above the expected value of BMI.

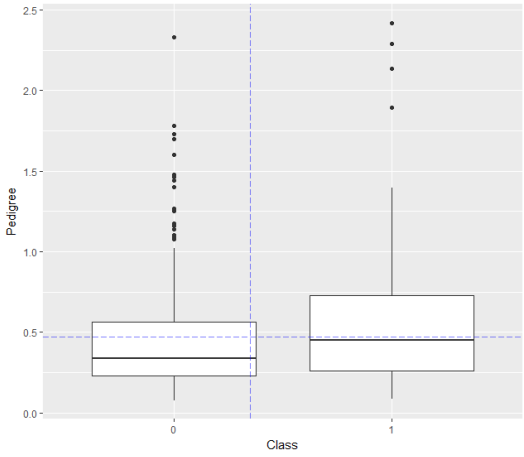

Figure 5.1.7: Boxplots of Pedigree per Class. The effect the variable Pedigree to the dependent variable is not significant.

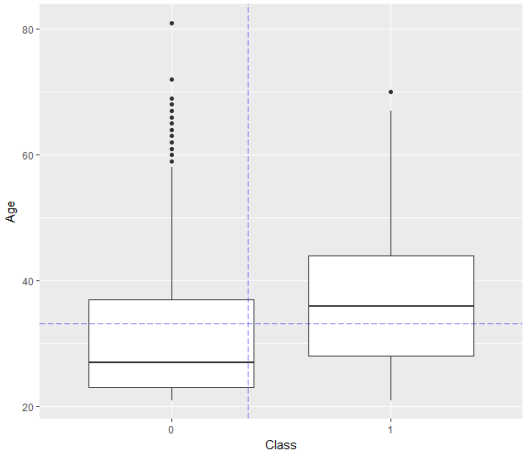

Figure 5.1.8: Boxplots of Age per Class. The medians of of the two classes differ a lot. Hence, we would say that age plays an important role in deciding if a woman is diabetic or no.

Observing the above boxplots we could say that the variables that seem to affect our dependent variable are pregnancy, glucose, BMI and age. This conclusion comes from the fact that the median of the first class of our dependent variable differs a lot from the median of the second class. Specifically, in all of the aforementioned variables the median of the first class is below the expected value of the variable while the median of the second class is above it. Pedigree seems to have a less significant effect while blood pressure, triceps and insulin do not seem to play an important role in changes to our dependent variable. We conclude to it by

https://doi.org/10.26219/heal.aueb.5951

observing that the medians of the two classes that correspond to those variables do not vary significantly.

5.2 Application of methods to the problem

Now, we are going to present the numerical results of this problem. As we mentioned above, we use Dantzig selector, Lasso and ols estimators in order to be able to compare our method with the others. We have to note that we quote two tables, each one of which contains the results of all methods in order to be able to compare the methods in an easy way. The only difference between the two tables is that the first one corresponds in the case when the starting values of Dantzig selector are random numbers while in the second case this does not happen. Let us take a look at the first case:

	Lasso		Ols method		Dantzig selector (random starting values)	
	Estimates (λ min)	Estimates (λ 1se)	Estimates	Standard errors	Estimates	Standard errors
Intercept	-0.8030	-0.4665	-0.8539	0.0855	-0.0677	0.0851
Pregnant	0.0191	0.0089	0.0206	0.0051	0.0049	0.0056
Glucose	0.0056	0.0048	0.0059	0.0005	0.0041	0.0006
Blood press	-0.0016	0.0000	-0.0023	0.0008	-0.0048	0.0008
Triceps	0.0000	0.0000	0.0002	0.0011	0.0003	-
Insulin	-0.0001	0.0000	-0.0002	0.0001	0.0001	-
BMI	0.0120	0.0064	0.0132	0.0021	0.0041	0.0021
Pedigree	0.1237	0.0025	0.1472	0.0451	-0.0243	0.0483
Age	0.0022	0.0000	0.0026	0.0015	0.0038	0.0017

Table 5.2.1: Table with the estimates of the three different methods (Lasso (two values of λ), ols estimators and Dantzig selector). Random numbers used as starting values in Dantzig selector.

Observing the results, we understand that estimations of coefficients by Dantzig selector differ a little from the estimations derived from the other methods. So, it can lead us to mistaken conclusions if we try to interpret the size of the influence of each variable. However, we could say that it detects the variables that do not affect the dependent and sets the estimations of their coefficients close to zero. It is more obvious if we look at the estimations of coefficients for the variables triceps and insulin. Both Lasso and ols method offer estimations close to zero for the coefficients of these two variables. These estimations are also given by Dantzig

https://doi.org/10.26219/heal.aueb.5951

selector, so we could say that it seems to work properly in detecting the variables that do not need to take part in our model even in the case of random starting values.

Below we present the results that correspond to the second case we mentioned above. In this case, we use the estimations of the ols method as starting values for the Dantzig selector. In this way, we try to facilitate Dantzig selector in its effort to estimate the coefficients.

	Lasso		Ols method		Dantzig selector (ols estimators as starting values)	
	Estimates (λ min)	Estimates (λ 1se)	Estimates	Standard errors	Estimates	Standard errors
Intercept	-0.8030	-0.4665	-0.8539	0.0855	-0.6682	0.0682
Pregnant	0.0191	0.0089	0.0206	0.0051	0.0223	0.0055
Glucose	0.0056	0.0048	0.0059	0.0005	0.0055	0.0005
Blood press	-0.0016	0.0000	-0.0023	0.0008	-0.0029	-
Triceps	0.0000	0.0000	0.0002	0.0011	0.0001	-
Insulin	-0.0001	0.0000	-0.002	0.0001	-0.0001	-
BMI	0.0120	0.0064	0.0132	0.0021	0.0114	0.0020
Pedigree	0.1237	0.0025	0.1472	0.0451	0.1405	0.0467
Age	0.0022	0.0000	0.0026	0.0015	0.0016	-

Table 5.2.2: Table with the estimates of the three different methods (Lasso (two values of λ), ols estimators and Dantzig selector). Ols estimators used as starting values in Dantzig selector.

Observing the results, we could say that we achieve our objective as the estimates are much better than the case of random starting values. Not only it detects the useless variables, but it also offers more accurate estimates of the coefficients for the rest of the variables. Hence, we could say that the procedure we followed above is an easy way of selecting starting values. In this way, we avoid selecting random starting values, which can lead to mistaken results.

Now, if we would like to comment on the standard errors of the coefficients, we could say that standard errors of Dantzig selector look like standard errors that derive from the ols method. That seems to happen even if we use random starting values. We do not present standard errors of all of the variables because the method we use for variance estimation computes standard errors only for the variables that are included in the model. Hence, if the estimation of a coefficient is close to zero, the corresponding variable is rejected by Dantzig selector and as a result, standard

https://doi.org/10.26219/heal.aueb.5951

error is not computed. However, if we would like to compute standard errors of these variables, we could suppose that these variables take part in the model too. The table below contains standard errors of all the coefficients, for both cases of starting values.

	Standard errors (ols method)	Standard errors (DS, random starting values)	Standard errors (DS, ols estimators as starting values)
Intercept	0.0855	0.0851	0.0682
Pregnant	0.0051	0.0056	0.0055
Glucose	0.0005	0.0020	0.0005
Blood press	0.0008	0.0008	0.0008
Triceps	0.0011	0.0012	0.0012
Insulin	0.0001	0.0002	0.0002
BMI	0.0021	0.0021	0.0020
Pedigree	0.0451	0.0483	0.0467
Age	0.0015	0.0017	0.0017

Table 5.2.3: Table which contains the standard errors of all the coefficients, for both the ols method and the method of Dantzig selector.

Observing the table 5.2.3, we could say that Dantzig selector gives almost the same standard errors as the ols method for both cases of starting values, but we have small deviations. Specifically, values of the standard errors that correspond to the Dantzig selector seem to be a little higher than those of the ols method. Small deviations are also observed between the two cases of starting values in Dantzig selector. When we use ols estimators as starting values, we observe small decrease at the standard errors of some of the variables. The above comments become more comprehensible if we take a look at the values of the standard errors of the variables pregnant, glucose, pedigree and age. Especially for the variable pedigree, we observe the value 0.0451 when we use the ols method and the values 0.0483 and 0.0467 when we use Dantzig selector with random and non-random starting values respectively.

The above problem does not refer to the case when the number of observations (n) is small relative to the number of predictors (p), therefore we studied it to observe how Dantzig selector works and compare its performance with Lasso in a real data example.

https://doi.org/10.26219/heal.aueb.5951

Chapter 6: Conclusions and discussion

The aim of the present study was the performance of Dantzig selector in the context of linear models. Firstly, we presented the theoretical background of Dantzig selector and subsequently, we tried to study some applications of the method. These applications concern either simulated data or real data examples. During the analysis, we also applied the method of Lasso to the aforementioned data in order to compare the two methods and come to useful conclusions for Dantzig selector.

Different combinations of the number of observations (n) and the number of predictors (p) were studied. We also used different values of the correlation (r) between the variables. Observing the estimations of the coefficients and the RMSEs, derived from the two methods, led us to useful conclusions. Specifically, we should mention that, in general, Lasso has better values of the RMSE than Dantzig selector. On the other hand, if we decrease the number of observations (n) and increase the number of predictors (p), Lasso has great difficulty in detecting which predictors should be included in the model. Dantzig selector seems to outperform Lasso in detecting the correct variables when $n<<p$, especially when there is high correlation between the variables. All of the above results derive from the case in which we use the true coefficients as starting values in Dantzig selector. This premise knowledge of the true coefficients, which is not the case in practice. Hence, we should mention that if we use random numbers as starting values in Dantzig selector, the arising estimators vary for different number of predictors. For difficult problems, random starting values result in disappointing estimations.

Concluding, we could say that if we use right starting values in Dantzig selector, estimations seem to be satisfactory. So, studies should may focus on finding a way of selecting proper starting values. A proposal in this direction could be splitting the sample into groups and applying the ols method in each one of them. Then, we could use ols estimators as starting values in Dantzig selector. As we noted in the real-world problem, this might facilitate Dantzig selector in its effort to give proper estimators.

https://doi.org/10.26219/heal.aueb.5951

Appendix

Clarifications

1) Although we presented in section 2 the theory of computing the variance in Dantzig selector, we did not quote the variance of the estimations in the simulation schemes presented in section 4. The reason is that for computing the variance we have to compute the quantities $(t(DM_2)\%*\%DM_2)^{-1}$ and $(t(DM_1)\%*\%DM_1)^{-1}$. The results of the computations $t(DM_2)\%*\%DM_2$ and $t(DM_1)\%*\%DM_1$ are matrices, which are not reversible when $n<<p$[13]. Hence, we cannot compute variance.

2) Correlation between the variables depends on the covariance matrix we use to produce the data. Hence, when we say that:

i) the correlation is r=0, then the covariance matrix is $\begin{pmatrix} 4 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 4 \end{pmatrix}$.

ii) the correlation is r=0.5, then the covariance matrix is $\begin{pmatrix} 4 & \cdots & 2 \\ \vdots & \ddots & \vdots \\ 2 & \cdots & 4 \end{pmatrix}$.

iii) the correlation is r=0.96, then the correlation matrix is $\begin{pmatrix} 52 & \cdots & 50 \\ \vdots & \ddots & \vdots \\ 50 & \cdots & 52 \end{pmatrix}$.

3) In section 3, we mentioned that we used the command nloptr. One of the arguments of nloptr refers to the algorithm that is used and one option of this argument is used as termination condition. This option is maxeval, which has a default value 1000, but we changed it to 20000. The reason we changed this value is that it did not let the algorithm to do the number of iterations it needed.

[13] DM_1 and DM_2 are defined in section 2.

https://doi.org/10.26219/heal.aueb.5951

Code in R

Now we are going to present the code that was used for the thesis with short description of what every command set does. We highlight with blue colour the commands of the algorithm that need changes if we want to change the correlation between the variables (r), the number of observations (n) or the true values of the coefficients.

Firstly, we quote the code used for the simple problem described in the 3rd section. These commands were used for the results of table 3.1:

```
library(nloptr)
library(MASS)
mu<-rep(5,3)
mu
Sigma<-matrix(c(4,2,2,2,4,2,2,2,4),3,3)
Sigma
x<-mvrnorm(n=100,mu=mu,Sigma=Sigma)
x
cov(x)
cor(x)
ones<-rep(1,100)
DM<-cbind(ones,x)
DM

nl<-matrix(rep(0,800),200,4)
ml<-matrix(rep(0,800),200,4)

for (i in 1:200){
e<-rnorm(100,0,sqrt(6))

y<-13+5*x[,1]+7*x[,2]+e
y

eval_f<-function(b){
return ("objective"=abs(b[1])+abs(b[2])+abs(b[3])+abs(b[4]))
}

eval_g_ineq <- function( b ) {
   constr <- c(abs(t(DM)%*%(y-(DM%*%c(b[1],b[2],b[3],b[4]))))-rep(0.001,4) )
   return("constraints"=constr)
}
```

https://doi.org/10.26219/heal.aueb.5951

```
res<- nloptr( x0=c(13,5,7,0),
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq,
              opts = list("algorithm"="NLOPT_LN_COBYLA","maxeval"   =
1000,"xtol_rel"=1.0e-4))
nl[i,]<-res$solution

model<-lm(y~x)
ml[i,]<-model$coef
}

apply(nl,2,mean)
```

Subsequently, we quote the code used for computing the variance and the confidence intervals referred in the 3rd section:

```
library(nloptr)
library(MASS)
mu<-rep(5,3)
mu
Sigma<-matrix(c(4,2,2,2,4,2,2,2,4),3,3)
Sigma
n<-100
x<-mvrnorm(n,mu=mu,Sigma=Sigma)
x
cov(x)
cor(x)
ones<-rep(1,n)
DM<-cbind(ones,x)
DM

a<-NULL
b<-NULL
c<-NULL
nl<-matrix(rep(0,800),200,4)

for (k in 1:200){
e<-rnorm(100,0,sqrt(6))

y<-13+5*x[,1]+7*x[,2]+e
y

h<-1:100
group1<-sample(h,50)
```

https://doi.org/10.26219/heal.aueb.5951

```
group1
group2<-h[-group1]
group2

ones_n2<-rep(1,n/2)
DM1<-cbind(ones_n2,x[group1,])
DM2<-cbind(ones_n2,x[group2,])

eval_f<-function(b){
return ("objective"=sum(abs(b[1:dim(x)[2]])))
}

eval_g_ineq_1 <- function( b ) {
   constr <- c(abs(t(DM1)%*%(y[group1]-(DM1%*%b[1:dim(DM1)[2]])))-rep(0.001,4) )
   return("constraints"=constr)
}

res_1<- nloptr( x0=c(13,5,7,0),
            eval_f=eval_f,
            eval_g_ineq = eval_g_ineq_1,
            opts = list("algorithm"="NLOPT_LN_COBYLA",
            "maxeval"   = 1000,"xtol_rel"=1.0e-4))

eval_g_ineq_2 <- function( b ) {
   constr <- c(abs(t(DM2)%*%(y[group2]-(DM2%*%b[1:dim(DM2)[2]])))-rep(0.001,4) )
   return("constraints"=constr)
}

res_2<- nloptr( x0=c(13,5,7,0),
            eval_f=eval_f,
            eval_g_ineq = eval_g_ineq_2,
            opts = list("algorithm"="NLOPT_LN_COBYLA",
            "maxeval"   = 1000,"xtol_rel"=1.0e-4))

m_1<-NULL
m_2<-NULL
for (i in 2:(length(res_1$solution))){
if (res_1$solution[i]>0.5) m_1<-c(m_1,i-1)
}

for (i in 2:length(res_2$solution)){
if (res_2$solution[i]>0.5) m_2<-c(m_2,i-1)
}
```

https://doi.org/10.26219/heal.aueb.5951

```
m<-NULL
for (i in 1:length(m_1)){
for (j in 1:length(m_2)){
if (m_1[i]==m_2[j]) m<-c(m,i)
}
}

I_n2<-matrix(rep(0,(n/2)^2),n/2,n/2)
diag(I_n2)<-rep(1,n/2)
s_1<-(t(y[group2])%*%(I_n2-
DM2[,c(1,m_1+1)]%*%solve(t(DM2[,c(1,m_1+1)])%*%DM2[,c(1,m_1+1)])%*%t(DM2[,c
(1,m_1+1)]))%*%y[group2])/((n/2)-length(m_1))
s_1

s_2<-(t(y[group1])%*%(I_n2-
DM1[,c(1,m_2+1)]%*%solve(t(DM1[,c(1,m_2+1)])%*%DM1[,c(1,m_2+1)])%*%t(DM1[,c
(1,m_2+1)]))%*%y[group1])/((n/2)-length(m_2))
s_2

s<-(s_1+s_2)/2
s

DMM<-DM[,c(1,m+1)]

eval_f_te<-function(b){
return ("objective"=sum(abs(b[c(1,m+1)])))
}

eval_g_ineq<- function( b ) {
   constr <- c(abs(t(DMM)%*%(y-(DMM%*%b[c(1,m+1)])))-rep(0.001,3) )
   return("constraints"=constr)
}

res<- nloptr( x0=c(13,5,7,0),
              eval_f=eval_f_te,
              eval_g_ineq = eval_g_ineq,
              opts = list("algorithm"="NLOPT_LN_COBYLA",
              "maxeval"   = 1000,"xtol_rel"=1.0e-4))

nl[k,]<-res$solution
z<-1.96*diag(sqrt((solve(t(DMM[group1,])%*%DMM[group1,])+
solve(t(DMM[group2,])%*%DMM[group2,]))*as.numeric(s/4)))
ci<-cbind(res$solution-z,res$solution+z)
if ((ci[1,1]<13)&(13<ci[1,2])) a<-c(a,1)
```

https://doi.org/10.26219/heal.aueb.5951

```
if ((ci[2,1]<5)&(5<ci[2,2]))  b<-c(b,1)
if ((ci[3,1]<7)&(5<ci[3,2]))   c<-c(c,1)
}
pososta<-c(sum(a)/200,sum(b)/200,sum(c)/200)
pososta

thetas<-c(13,5,7,0)
j<-matrix(rep(0,600),200,3)
for (i in 1:4){
j[,i]<-(nl[,i]-rep(thetas[i],200))^2
}
pos<-apply(j,2,sum)
rmse<-sqrt(pos/200)
stand_errors<-z/1.96
summary(model)
```

Now, we quote the code used for the more complicated simulation schemes, which are presented in the 4th section. We present one command set for the Dantzig selector and one command set for the lasso. For various combinations of n, λ and r we only change a few numbers each time. We put blue color to those commands that need to change from time to time. This command set refers to the Dantzig selector:

```
library(nloptr)
library(MASS)
mu<-rep(5,100)
Sigma<-matrix(rep(0,10000),100,100)
diag(Sigma)<-rep(4,100)
n<-25
x<-mvrnorm(n,mu=mu,Sigma=Sigma)
x
ones<-rep(1,n)
DM<-cbind(ones,x)
nl<-matrix(rep(0,50500),500,101)

j<-NULL
p<-c(13,5,7,8,6,9,5,7,7,6,4,rep(0,90))

for (i in 1:500){
e<-rnorm(n,0,sqrt(6))

y<-13+5*x[,1]+7*x[,2]+8*x[,3]+6*x[,4]+9*x[,5]+5*x[,6]+7*x[,8]+7*x[,9]+6*x[,10]+
4*x[,11] +e
y
```

https://doi.org/10.26219/heal.aueb.5951

```
eval_f<-function(b){
return ("objective"=sum(abs(b[1:dim(DM)[2]])))
}

eval_g_ineq <- function( b ) {
    constr <- c(abs(t(DM)%*%(y-(DM%*%b[1:dim(DM)[2]])))-rep(0.001,dim(DM)[2]) )
   return("constraints"=constr)
}

res<- nloptr( x0=c(13,5,7,8,6,9,5,7,7,6,4,rep(0,90)),
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq,
              opts = list("algorithm"="NLOPT_LN_COBYLA","maxeval"   =
20000,"xtol_rel"=1.0e-4))
nl[i,]<-res$solution

}
apply(nl,2,mean)
thetas<-c(13,5,7,8,6,9,5,7,7,6,4,rep(0,90))
j<-matrix(rep(0,50500),500,101)
for (i in 1:101){
j[,i]<-(nl[,i]-rep(thetas[i],500))^2
}
pos<-apply(j,2,sum)
rmse<-sqrt(pos/500)
```

Then, we quote the code that corresponds to the Lasso:

```
library(nloptr)
library(MASS)
mu<-rep(5,100)
Sigma<-matrix(rep(0,10000),100,100)
diag(Sigma)<-rep(4,100)
n<-10
x<-mvrnorm(n,mu=mu,Sigma=Sigma)
x
ones<-rep(1,n)
DM<-cbind(ones,x)
lasso_min<-matrix(rep(0,50500),500,101)
lasso_1se<-matrix(rep(0,50500),500,101)

j<-NULL
p<-c(13,5,7,8,6,9,5,7,7,6,4,rep(0,90))

for (i in 1:500){
e<-rnorm(n,0,sqrt(6))
```

https://doi.org/10.26219/heal.aueb.5951

```
y<-13+5*x[,1]+7*x[,2]+8*x[,3]+6*x[,4]+9*x[,5]+5*x[,6]+7*x[,8]+7*x[,9]+6*x[,10]+
4*x[,11]+e
y

library(glmnet)
lasso<-cv.glmnet(x,y)
lasso$lambda.min
lasso$lambda.1se
blasso<-coef(lasso,s="lambda.min")
blasso
lasso_min[i,]<-blasso[,1]

blassod<-coef(lasso,s="lambda.1se")
blassod
lasso_1se[i,]<-blassod[,1]

}
apply(lasso_min,2,mean)

thetas<-c(13,5,7,8,6,9,5,7,7,6,4,rep(0,90))
j<-matrix(rep(0,50500),500,101)
for (i in 1:101){
j[,i]<-(lasso_min[,i]-rep(thetas[i],500))^2
}
Pos_min<-apply(j,2,sum)
rmse_min<-sqrt(pos_min/500)

apply(lasso_1se,2,mean)

j<-matrix(rep(0,50500),500,101)
for (i in 1:101){
j[,i]<-(lasso_1se[,i]-rep(thetas[i],500))^2
}
pos_1se<-apply(j,2,sum)
rmse_1se<-sqrt(pos_1se/500)
```

Here, we show the code used for extracting the results in Table 4.1.3 in which we also present the standard errors of the coefficients in order to compare them with the corresponding RMSEs.

```
library(nloptr)
library(MASS)
mu<-rep(5,10)
Sigma<-matrix(rep(2,100),10,10)
diag(Sigma)<-rep(4,10)
n<-100
```

https://doi.org/10.26219/heal.aueb.5951

```
x<-mvrnorm(n,mu=mu,Sigma=Sigma)
x
ones<-rep(1,n)
DM<-cbind(ones,x)
nl<-matrix(rep(0,5500),500,11)

a<-NULL
b<-NULL
c<-NULL
j<-NULL
variance<-NULL

for (i in 1:500){
e<-rnorm(n,0,sqrt(6))

y<-13+5*x[,1]+7*x[,2]+e
y

eval_f<-function(b){
return ("objective"=sum(abs(b[1:dim(DM)[2]])))
}

eval_g_ineq <- function( b ) {
    constr <- c(abs(t(DM)%*%(y-(DM%*%b[1:dim(DM)[2]])))-rep(0.001,dim(DM)[2]) )
   return("constraints"=constr)
}

res<- nloptr( x0=c(13,5,7,0,0,0,0,0,0,0,0),
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq,
              opts = list("algorithm"="NLOPT_LN_COBYLA","maxeval"   =
5000,"xtol_rel"=1.0e-4))
nl[i,]<-res$solution

h<-1:n
group1<-sample(h,n/2)
group1
group2<-h[-group1]
group2

ones_n2<-rep(1,n/2)
DM1<-cbind(ones_n2,x[group1,])
DM2<-cbind(ones_n2,x[group2,])

eval_g_ineq_1 <- function( b ) {
   constr <- c(abs(t(DM1)%*%(y[group1]-(DM1%*%b[1:dim(DM1)[2]])))-
rep(0.001,dim(DM1)[2]) )
```

https://doi.org/10.26219/heal.aueb.5951

```
   return("constraints"=constr)
}

res_1<- nloptr( x0=c(13,5,7,0,0,0,0,0,0,0,0),
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq_1,
              opts = list("algorithm"="NLOPT_LN_COBYLA",
              "maxeval"   = 5000,"xtol_rel"=1.0e-4))

eval_g_ineq_2 <- function( b ) {
   constr <- c(abs(t(DM2)%*%(y[group2]-(DM2%*%b[1:dim(DM2)[2]])))-
rep(0.001,dim(DM2)[2]) )
   return("constraints"=constr)
}

res_2<- nloptr( x0=c(13,5,7,0,0,0,0,0,0,0,0),
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq_2,
              opts = list("algorithm"="NLOPT_LN_COBYLA",
              "maxeval"   = 5000,"xtol_rel"=1.0e-4))

m_1<-NULL
m_2<-NULL
for (i in 2:(length(res_1$solution))){
if (res_1$solution[i]>0.5) m_1<-c(m_1,i-1)
}

for (i in 2:length(res_2$solution)){
if (res_2$solution[i]>0.5) m_2<-c(m_2,i-1)
}

m<-NULL
for (i in 1:length(m_1)){
for (j in 1:length(m_2)){
if (m_1[i]==m_2[j]) m<-c(m,m_1[i])
}
}

I_n2<-matrix(rep(0,(n/2)^2),n/2,n/2)
diag(I_n2)<-rep(1,n/2)
s_1<-(t(y[group2])%*%(I_n2-
DM2[,c(1,m_1+1)]%*%solve(t(DM2[,c(1,m_1+1)])%*%DM2[,c(1,m_1+1)])%*%t(DM2[,c
(1,m_1+1)]))%*%y[group2])/((n/2)-length(m_1))
s_1
```

https://doi.org/10.26219/heal.aueb.5951

```
s_2<-(t(y[group1])%*%(I_n2-
DM1[,c(1,m_2+1)]%*%solve(t(DM1[,c(1,m_2+1)])%*%DM1[,c(1,m_2+1)])%*%t(DM1[,c
(1,m_2+1)]))%*%y[group1])/((n/2)-length(m_2))
s_2

s<-(s_1+s_2)/2
s

DMM<-DM[,c(1,m+1)]

eval_f_te<-function(b){
return ("objective"=sum(abs(b[c(1,m+1)])))
}

eval_g_ineq_va<- function( b ) {
   constr <- c(abs(t(DMM)%*%(y-(DMM%*%b[c(1,m+1)])))-
rep(0.001,length(b[c(1,m+1)])) )
   return("constraints"=constr)
}

res_va<- nloptr( x0=c(13,5,7,0,0,0,0,0,0,0,0),
            eval_f=eval_f_te,
            eval_g_ineq = eval_g_ineq_va,
            opts = list("algorithm"="NLOPT_LN_COBYLA",
            "maxeval"   = 5000,"xtol_rel"=1.0e-4))

z<-1.96*diag(sqrt((solve(t(DMM[group1,])%*%DMM[group1,])+
solve(t(DMM[group2,])%*%DMM[group2,]))*as.numeric(s/4)))
ci<-cbind(res_va$solution-z,res_va$solution+z)
if ((ci[1,1]<13)&(13<ci[1,2])) a<-c(a,1)
if ((ci[2,1]<5)&(5<ci[2,2]))  b<-c(b,1)
if ((ci[3,1]<7)&(5<ci[3,2]))   c<-c(c,1)
variance<-
rbind(variance,diag((solve(t(DMM[group1,])%*%DMM[group1,])+solve(t(DMM[group2,
])%*%DMM[group2,]))*as.numeric(s/4)))

}
apply(nl,2,mean)
pososta<-c(sum(a)/500,sum(b)/500,sum(c)/500)
pososta
apply(variance,2,mean)
thetas<-c(13,5,7,0,0,0,0,0,0,0,0)
j<-matrix(rep(0,5500),500,11)
for (i in 1:11){
j[,i]<-(nl[,i]-rep(thetas[i],500))^2
}
pos<-apply(j,2,sum)
```

https://doi.org/10.26219/heal.aueb.5951

```
rmse<-sqrt(pos/500)
rmse
stand_errors<-z/1.96
```

Finally, we present the code used to the real-world problem. We used the below commands for the boxplots we made:

```
diabetes<-read.table("http://anson.ucdavis.edu/~jihao/pima-indians-diabetes.dat",
sep=",")
colnames(diabetes)<-c("pregnant",
"glucose","blood.press", "triceps", "insulin",
"BMI", "pedigree", "age", "class")
diabetes

library(plotrix)
table(diabetes$class)
x<-table(diabetes$class)
barp(x,col=5,cylindrical=T,shadow=F,names.arg=c("0","1"),xlab="Class",ylab="Frequenci
es")

library(Hmisc)
library(grid)
library(lattice)
library(survival)
library(Formula)
library(ggplot2)

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$pregnant))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$pregnant),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Pregnant")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$glucose))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$glucose),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Glucose")
p
```

https://doi.org/10.26219/heal.aueb.5951

```
p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$blood.press))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$blood.press),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Blood press")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$triceps))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$triceps),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Triceps")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$insulin))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$insulin),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Insulin")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$BMI))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$BMI),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("BMI")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$pedigree))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$pedigree),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
```

https://doi.org/10.26219/heal.aueb.5951

```
p<-p+ylab("Pedigree")
p

p<-ggplot(data=diabetes,aes(x=as.factor(diabetes$class),y=diabetes$age))
p<-p+geom_boxplot()
p<-p+geom_hline(show_guide=T,yintercept=mean(diabetes$age),
linetype="longdash",alpha=0.5,color="blue")
p<-p+geom_vline(xintercept=mean(diabetes$class)-
min(diabetes$class)+1,linetype="longdash",color="blue",alpha=0.5)
p<-p+xlab("Class")
p<-p+ylab("Age")
p
```

Subsequently, we present the code used for the application of the different methods (ols, lasso, dantzig) to our data:

```
diabetes<-read.table("http://anson.ucdavis.edu/~jihao/pima-indians-diabetes.dat",
sep=",")
colnames(diabetes)<-c("pregnant",
"glucose","blood.press", "triceps", "insulin",
"BMI", "pedigree", "age", "class")

library(nloptr)
library(MASS)
x<-diabetes[,-9]
x<-as.matrix(x)
ones<-rep(1,dim(x)[1])
DM<-cbind(ones,x)
y<-as.numeric(diabetes[,9])
DM<-as.matrix(DM)
n<-dim(diabetes)[1]
model<-lm(y~x)
model$coef

eval_f<-function(b){
return ("objective"=sum(abs(b[1:dim(x)[2]])))
}

eval_g_ineq <- function( b ) {
    constr <- c(abs(t(DM)%*%(y-(DM%*%b[1:dim(DM)[2]])))-rep(0.001,dim(DM)[2]) )
   return("constraints"=constr)
}

res<- nloptr( x0=model$coef,
              eval_f=eval_f,
              eval_g_ineq = eval_g_ineq,
```

https://doi.org/10.26219/heal.aueb.5951

```
            opts = list("algorithm"="NLOPT_LN_COBYLA","maxeval"   =
5000,"xtol_rel"=1.0e-4))

h<-1:n
group1<-sample(h,n/2)
group1
group2<-h[-group1]
group2
ones_n2<-rep(1,n/2)
DM1<-cbind(ones_n2,x[group1,])
DM2<-cbind(ones_n2,x[group2,])

eval_g_ineq_1 <- function( b ) {
   constr <- c(abs(t(DM1)%*%(y[group1]-(DM1%*%b[1:dim(DM1)[2]])))-
rep(0.001,dim(DM1)[2]) )
   return("constraints"=constr)
}

res_1<- nloptr( x0=model$coef,
            eval_f=eval_f,
            eval_g_ineq = eval_g_ineq_1,
            opts = list("algorithm"="NLOPT_LN_COBYLA",
            "maxeval"   = 5000,"xtol_rel"=1.0e-4))

eval_g_ineq_2 <- function( b ) {
   constr <- c(abs(t(DM2)%*%(y[group2]-(DM2%*%b[1:dim(DM2)[2]])))-
rep(0.001,dim(DM2)[2]) )
   return("constraints"=constr)
}

res_2<- nloptr( x0=model$coef,
            eval_f=eval_f,
            eval_g_ineq = eval_g_ineq_2,
            opts = list("algorithm"="NLOPT_LN_COBYLA",
            "maxeval"   = 5000,"xtol_rel"=1.0e-4))

m_1<-NULL
m_2<-NULL
for (i in 2:(length(res_1$solution))){
if (abs(res_1$solution[i])>0.005) m_1<-c(m_1,i-1)
}

for (i in 2:length(res_2$solution)){
if (abs(res_2$solution[i])>0.005) m_2<-c(m_2,i-1)
}

m<-NULL
```

https://doi.org/10.26219/heal.aueb.5951

```
for (i in 1:length(m_1)){
for (j in 1:length(m_2)){
if (m_1[i]==m_2[j]) m<-c(m,m_1[i])
}
}

I_n2<-matrix(rep(0,(n/2)^2),n/2,n/2)
diag(I_n2)<-rep(1,n/2)
s_1<-(t(y[group2])%*%(I_n2-
DM2[,c(1,m_1+1)]%*%solve(t(DM2[,c(1,m_1+1)])%*%DM2[,c(1,m_1+1)])%*%t(DM2[,c
(1,m_1+1)]))%*%y[group2])/((n/2)-length(m_1))
s_1

s_2<-(t(y[group1])%*%(I_n2-
DM1[,c(1,m_2+1)]%*%solve(t(DM1[,c(1,m_2+1)])%*%DM1[,c(1,m_2+1)])%*%t(DM1[,c
(1,m_2+1)]))%*%y[group1])/((n/2)-length(m_2))
s_2

s<-(s_1+s_2)/2
s

DMM<-DM[,c(1,m+1)]

eval_f_te<-function(b){
return ("objective"=sum(abs(b[c(1,m+1)])))
}

eval_g_ineq_va<- function( b ) {
   constr <- c(abs(t(DMM)%*%(y-(DMM%*%b[c(1,m+1)])))-
rep(0.001,length(b[c(1,m+1)])) )
   return("constraints"=constr)
}

res$solution
z<-1.96*diag(sqrt((solve(t(DMM[group1,])%*%DMM[group1,])
+solve(t(DMM[group2,])%*%DMM[group2,]))*as.numeric(s/4)))
stand_errors<-z/1.96

summary(model)

diabetes<-read.table("http://anson.ucdavis.edu/~jihao/pima-indians-diabetes.dat",
sep=",")
colnames(diabetes)<-c("pregnant",
"glucose","blood.press", "triceps", "insulin",
"BMI", "pedigree", "age", "class")

x<-diabetes[,-9]
```

https://doi.org/10.26219/heal.aueb.5951

```
x<-as.matrix(x)
ones<-rep(1,dim(x)[1])
DM<-cbind(ones,x)
y<-diabetes[,9]
DM<-as.matrix(DM)

library(glmnet)
lasso<-cv.glmnet(x,y)
lasso$lambda.min
lasso$lambda.1se
blasso<-coef(lasso,s="lambda.min")
blasso

blassod<-coef(lasso,s="lambda.1se")
blassod

model<-lm(y~x)
model$coef

cbind(round(blasso[,1],4),round(blassod[,1],4),round(res$solution,4),round(model$coe
f,4))
```

https://doi.org/10.26219/heal.aueb.5951

Bibliography-references

Papers

- EMMANUEL CANDES AND TERENCE TAO, “The Dantzig selector: Statistical estimation when p is much larger than n”, The Annals of Statistics (2007)
- ROBERT TIBSHIRANI, “Regression Shrinkage and Selection via the Lasso”, Journal of the Royal Statistical Society (1996)
- JIANQING FAN, SHAOJUN GUO AND NING HAO, “Variance estimation using refitted cross-validation in ultrahigh dimensional regression”, Journal of the Royal Statistical Society (2012)
- EMMANUEL CANDES AND TERENCE TAO, “Decoding by Linear Programming”, IEEE Trans. Inform. Theory (2005)
- GARETH M. JAMES AND PETER RADCHENKO, “A generalized Dantzig selector with shrinkage tuning”, Biometrika (2009)
- N. MEINSHAUSEN, G. ROCHA AND B. YU, “Discussion: A tale of three cousins: Lasso, L2boosting and Dantzig”, The Annals of Statistics (2007)
- PETER RADCHENKO AND GARETH M. JAMES, “Variable Inclusion and Shrinkage Algorithms”, Marshall School of Business, University of Southern California (2008)
- HUI ZOU, “The Adaptive Lasso and Its Oracle Properties”, American Statistical Association, Journal of the American Statistical Association (2006)
- JIANQING FAN AND RUNZE LI, “Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties”, American Statistical Association, Journal of the American Statistical Association (2001)
- HUI ZOU AND TREVOR HASTIE, “Regularization and variable selection via the elastic net”, Journal of the Royal Statistical Society, Standford University, USA (2005)
- NICOLAI MEINSHAUSEN, “Relaxed Lasso”, Journal: Computational Statistics and Data Analysis (2006)

https://doi.org/10.26219/heal.aueb.5951

Books

- TREVOR HASTIE, ROBERT TIBSHIRANI AND JEROME FRIEDMAN, "The Elements of Statistical Learning", Springer Series in Statistics (Second edition, 2008)

Manuals

- Package 'nloptr', programming language R
- Package 'glmnet', programming language R

Lectures

- M. J. D POWELL, "A view of algorithms for optimization without derivatives", Presented as a William Benter Distinguished Lecture at the City University of Hong Kong (2007)

https://doi.org/10.26219/heal.aueb.5951

Index of tables and figures

https://doi.org/10.26219/heal.aueb.5951

- Figures
  - Complicated simulation schemes:
    - 10 predictors – 8 zero coefficients:
      - n=100 and r=0: Figures 4.1.1, 4.1.2, 4.1.3, 19, 20
      - n=100 and r=0.5: Figures 4.1.4, 4.1.5, 4.1.6, 21, 22
      - n=100 and r=0.96: Figures 4.1.7, 4.1.8, 4.1.9, 23, 24
      - n=20 and r=0: Figures 4.1.10, 4.1.11, 4.1.12, 25
      - n=20 and r=0.5: Figures 4.1.13, 4.1.14, 4.1.15, 26, 27
      - n=20 and r=0.96: Figures 4.1.16, 4.1.17, 4.1.18, 28, 29
      - n=7 and r=0: Figures 4.1.19, 4.1.20, 4.1.21, 29, 30
      - n=7 and r=0.5: Figures 4.1.22, 4.1.23, 4.1.24, 31, 32
      - n=7 and r=0.96: Figures 4.1.25, 4.1.26, 4.1.27, 32, 33
  - Real-world problem:
    - Pregnant vs Class: Figure5.1.1, 66
    - Glucose vs Class: Figure 5.1.2, 66
    - Blood pressure vs Class: Figure 5.1.3, 66
    - Triceps vs Class: Figure 5.1.4, 66
    - Insulin vs Class: Figure 5.1.5, 67
    - BMI vs Class: Figure 5.1.6, 67
    - Pedigree vs Class: Figure5.1.7, 67
    - Age vs Class: Figure 5.1.8, 67

https://doi.org/10.26219/heal.aueb.5951