Pontifícia Universidade Católica
do Rio de Janeiro

Daniel Martins Coutinho

A theory based, data driven selection for the regularization parameter for LASSO

Dissertação de Mestrado

Thesis presented to the Programa de Pós–graduação em Economia, do Departamento de Economia da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Economia.

Advisor: Prof. Marcelo Cunha Medeiros

Rio de Janeiro
November 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Daniel Martins Coutinho

A theory based, data driven selection for the regularization parameter for LASSO

Thesis presented to the Programa de Pós–graduação em Economia da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Economia. Approved by the Examination Committee:

Prof. Marcelo Cunha Medeiros
Advisor
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Prof. Ricardo Pereira Masini
FGV-SP

Prof. Anders Bredahl Koch
Ox

Rio de Janeiro, November the 6th, 2020

Daniel Martins Coutinho

Graduação em Ciências Econômicas pela PUC-Rio e mestrado em Ciências Econômicas pela PUC-Rio

Bibliographic data

Martins Coutinho, Daniel

A theory based, data driven selection for the regularization parameter for LASSO / Daniel Martins Coutinho; advisor: Marcelo Cunha Medeiros. – 2020.

38 f: il. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Economia, 2020.

Inclui bibliografia

1. Economia – Teses. 2. Aprendizado por Máquina. 3. LASSO. 4. adaLASSO. 5. Parâmetro de Regularização. I. Cunha Medeiros, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Economia. III. Título.

CDD: 000

Acknowledgments

I thank Marcelo Medeiros, my advisor, for the support and ideas. I would also like to thank my family, for their support in those two years. I have been fortunate to have many friends along this journey, and I thank, among others, Daniel Sá Earp, Leila Vieira and Lucas Maynard. The staff of the Economics Department at PUC-Rio were always helpful to solve the bureaucracy. I would also like to acknowledge the financial support by CNPq and CAPES.

Abstract

Martins Coutinho, Daniel; Cunha Medeiros, Marcelo (Advisor). **A theory based, data driven selection for the regularization parameter for LASSO**. Rio de Janeiro, 2020. 38p. Dissertação de Mestrado – Departamento de Economia, Pontifícia Universidade Católica do Rio de Janeiro.

We provide a new way to select the regularization parameter for the LASSO and adaLASSO. It is based on the theory and incorporates an estimate of the variance of the noise. We show theoretical properties of the procedure and Monte Carlo simulations showing that it is able to handle more variables in the active set than other popular options for the regularization parameter.

Keywords

Machine Learning; LASSO; adaLASSO; Regularization Parameter.

Resumo

Martins Coutinho, Daniel; Cunha Medeiros, Marcelo. **Selecionando o parâmetro de regularização para o LASSO: baseado na teoria e nos dados**. Rio de Janeiro, 2020. 38p. Dissertação de Mestrado – Departamento de Economia, Pontifícia Universidade Católica do Rio de Janeiro.

O presente trabalho apresenta uma nova forma de selecionar o parâmetro de regularização do LASSO e do adaLASSO. Ela é baseada na teoria e incorpora a estimativa da variância do ruído. Nós mostramos propriedades teóricas e simulações Monte Carlo que o nosso procedimento é capaz de lidar com mais variáveis no conjunto ativo do que outras opções populares para a escolha do parâmetro de regularização

Palavras-chave

Aprendizado por Máquina; LASSO; adaLASSO; Parâmetro de Regularização.

Table of contents

1 Introduction **10**
1.1 Notation 11

2 The Algorithm **12**

3 Theory **15**
3.1 Convergence of the Algorithm 16
3.2 Regularization Parameter 20

4 Simulations **21**
4.1 Convergence 21
4.2 Model selection 23
4.3 Regularization Parameter 27

5 Empirical Example **31**

6 Conclusion **33**

Bibliography **34**

List of figures

Figure 4.1 Experiment 1. Boxplot of estimated variances 22
Figure 4.2 Experiment 2. Boxplot of estimated variances 23
Figure 4.3 Experiment 3. Boxplot of estimated variances 23
Figure 4.4 Experiment 4. Boxplot of estimated variances 24
Figure 4.5 Experiment 5. Boxplot of estimated variances 24
Figure 4.6 Boxplot of estimated standard deviation 29
Figure 4.7 Boxplot of estimated standard deviation 30

List of tables

Table 4.1 Design 1: Result for 5000 replications 25
Table 4.2 Design 2: Result for 3000 replications 25
Table 4.3 Design 3: Result for 1000 replications 25
Table 4.4 Design 4: Result for 1000 replications 25
Table 4.5 Design 5: Result for 1000 replications 26
Table 4.6 Design 6: Result for 1000 replications 26
Table 4.7 Simulations with fixed design 26
Table 4.8 Simulations with Subexponential error 27
Table 4.9 Simulations with Polynomial Tails: Student's t Distribution with 4 degrees of freedom 27
Table 4.10 Simulations with Polynomial Tails: Student's t Distribution with 8 degrees of freedom 28
Table 4.11 Simulations with Polynomial Tails: Student's t Distribution with 3 degrees of freedom 28
Table 4.12 Design 1, 2000 replications 28
Table 4.13 Design 3, 2000 replications 28

Table 5.1 Coefficients: Effect over Violent Crimes 31
Table 5.2 Coefficients: Effect over Property Crimes 31
Table 5.3 Coefficients: Effect over Murder 31
Table 5.4 Number of variables selected: Violent Crimes 32
Table 5.5 Number of variables selected: Property Crimes 32
Table 5.6 Number of variables selected: Murder 32

1
Introduction

The linear model, usually estimated by ordinary least squares (OLS), is the workhorse for the analysis of economic data. It provides reliable statistical properties and easy interpretation. However, nowadays is not uncommon to have more variables than observations, which precludes the use of OLS. This arises in forecasting, in which one uses a large number of inputs to make better forecasts or when doing causal inference, in which one has a large number of variables that are potential confounders that should be used as controls. The LASSO (Least Absolute Shrinkage and Select Operator), first suggested by Tibshirani (1996), extends the usual OLS estimators and allows for more variables than observations. It is able to select variables using the ℓ_1 norm as a penalty, which induces kinks in the objective function.

The main issue with LASSO is selecting the regularization parameter. There are lots of possible choices: information criteria, Cross Validation and some attempts to choose the regularization parameter using the theory created for the LASSO. As shown in Bickel et al. (2009), and discussed in Bühlmann & Van de Geer (2011) and Wainwright (2019), setting the regularization parameter $\lambda = 2\sigma\sqrt{2\log(p)/n}$ guarantees good results. However, it requires knowledge of the variance of the error.

The contribution of this paper is twofold: we show how to use the regularization parameter in Bickel et al. (2009) using an iterative procedure in which at each step we estimate the model and, using the coefficients obtained, we compute the variance of the residual. We discuss the convergence properties of our algorithm. In general, it is not true that the LASSO version converges: it is highly dependent on the sample size, number of variables and the size of the active set. On the other hand, a simple twist of the LASSO, called the adaptive LASSO, first suggested by Zou (2006), allows us to have much better results regarding convergence.

The second contribution is to show that, when using the adaLASSO, one can use $\lambda = \sigma\sqrt{2\log(p)/n}$ and the proofs still work. We show theoretical results for this regularization parameter for the adaLASSO.

Bickel et al. (2009) is the main article in which we base our ideas. The selection of the regularization parameter is a key problem for the LASSO and

an active area of research. There are suggestions based on information criteria, as Zhang et al. (2010), Fan & Tang (2013) and Hui et al. (2015), among others; and some suggestions based on the theory, as Belloni et al. (2012) and Belloni et al. (2013). See Coutinho et al. (2017) for a review of different choices of the regularization parameter for the adaLASSO and the LASSO.

This paper has five sections: the next one describes the algorithm. Section 3 shows a bit of the theory. Section 4 shows the Monte Carlo simulations. The last section concludes.

1.1 Notation

We will say that $\hat{\beta}$ is the estimated vector of coefficients, and β^0 is the true vector of coefficients. X_S are the columns of X for which β^0 is different of zero, and X_{S^c} is the set for which the columns of X are equal to zero. Therefore, $\beta^0_S \neq 0$ and $\beta^0_{S^c} = 0$. There are p variables with s is the cardinality of the set S. We use $\|x_i\|_q = (\sum_i |x_i|^q)^{1/q}$, with the convention that $\|x_i\|_\infty = \max_i |x_i|$.

In the algorithm definition, we use $Sd(u) = \frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})$, the empirical standard deviation of u, in which $\bar{u}$ is the mean of u.

2
The Algorithm

Formally, the LASSO solves:

$$\beta_{LASSO} \in \arg\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda, \sum_{j=1}^{p} |\beta_j|,$$

in which λ is the regularization parameter. Algorithms for solving the LASSO problem are well establish and our algorithm focus in selecting the regularization parameter.

The choice of the regularization parameter is an important part of the algorithm, as it controls how many variables will be added to the model and the amount of shrinkage of the coefficients. Cross Validation is a popular choice (see Hastie et al. (2009)). However, Cross Validation might be slow for large data sets and is not suitable for dependent data without modifications. Information Criteria are also a possibility. Although one could use AIC and BIC, neither of these criteria were created having in mind a high dimensional setting. There are criteria created for the high dimensional case, as Zhang et al. (2010), Fan & Tang (2013) and Hui et al. (2015). Bickel et al. (2009) is one of the first articles suggesting a regularization parameter based on the theory and Bühlmann & Van de Geer (2011) explicit use the theory to suggest a feasible regularization parameter, using the variance of the dependent variable. Belloni et al. (2012) and Belloni et al. (2013) also provides a way to select the regularization parameter based on the theory, that handles heterocedasticity. This huge variety of procedures is due to the fact that they try to solve different problems and work for different kinds of DGPs.

Based on Bickel et al. (2009), we suggest that $\lambda = \sigma A\sqrt{2\log(p)/n}$, where σ is the standard deviation of the error, p is the number of regressors, potentially $p \gg n$, and A is a parameter - in Bickel et al. (2009), $A = 2$. This regularization parameter is unfeasible, since it depends on the standard deviation of the error. However, we can use an estimator of the standard deviation of the error, which we denote by $\hat{\sigma}$, to get a feasible version of the regularization parameter, $\hat{\lambda}$. We propose Algorithm 1 that uses the LASSO to generate the estimate for $\hat{\sigma}$ and iterates on it to get $\hat{\beta}$: start with a guess for the standard deviation and compute the LASSO using the regularization

parameter by Bickel et al. (2009). This will give us a vector of coefficients. Use this vector to calculate the residuals $\hat{u} := Y - X\beta$. Use the standard deviation of $\hat{u}$ as a new guess and iterate.

Input: Some guess for σ, the data
while *convergence fails* **do**
  1.Set $\hat{\lambda} = \hat{\sigma}\sqrt{2\log(p)/n}$;
  2.Estimate the LASSO ($\hat{\beta}_{LASSO}$) using $\hat{\lambda}$ as the regularization parameter;
  3.Compute the residual $\hat{u} = Y - X\hat{\beta}_{LASSO}$;
  4.Compute $\hat{\sigma} = Sd(\hat{u})$;
  if *convergence* **then**
    Return $\hat{\beta}_{LASSO}$ and report sucess
  else
    Go back to step 1
  end
end

Algorithm 1: An algorithm for LASSO

There can be several ways to define convergence. In the following simulations we use one of the two bellow:

1. The $\max(abs(\hat{\beta}_i - \hat{\beta}_{i-1}))$ is smaller than a δ
2. The $abs(\hat{\sigma}_i - \hat{\sigma}_{i-1})$ is smaller than a ε

i is the number of iteration and $abs(\cdot)$ stands for the absolute value function. We also limit the maximum number of iterations, so even if there is not convergence the algorithm still quits. It is simple to extended our algorithm to deal with adaptive LASSO (adaLASSO), Elastic Net or Thresholded LASSO. The adaLASSO uses a two step procedure in which we first estimate the model with the LASSO. In the second step, we solve the following problem:

$$\beta_{adaLASSO} \in \arg\min_{\beta} \frac{1}{2n}\sum_{i=1}^{n}(Y_i - X_i\beta)^2 + \lambda\sum_{j=1}^{p}\omega_j|\beta_j|,$$

in which ω_j are a set of positive weights. We implement two versions of the adaLASSO, and they only change with respect to the weights. The first version use fixed weights based on a first stage LASSO with the initial guess for the regularization parameter, so $\omega = 1/(1/\sqrt{n}+|\hat{\beta}_{LASSO}|)$, in which n is the sample size. This weight never changed again. The second version uses as weights the previous estimation of the adaLASSO, so $\omega = 1/(1/\sqrt{n} + |\hat{\beta}_{i-1}|)$, so the weights are updated at each step. The algorithm for the adaLASSO is shown in Algorithm 2.

Input: Some guess for σ ($\hat{\sigma}$), the data
a.Set $\hat{\lambda} = \hat{\sigma}\sqrt{2\log(p)/n}$;
b.Estimate the LASSO ($\hat{\beta}_{LASSO}$) using $\hat{\lambda}$ as the regularization parameter;
c. Set $\omega = \frac{1}{1/n+|\beta_{LASSO}|}$;
while *convergence fails* **do**
  1.Set $\hat{\lambda} = \hat{\sigma}\sqrt{2\log(p)/n}$;
  2.Estimate the adaLASSO ($\hat{\beta}_{LASSO}$) using $\hat{\lambda}$ as the regularization parameter and ω as weights;
  3.Compute the residual $\hat{u} = Y - X\hat{\beta}_{LASSO}$;
  4.Compute $\hat{\sigma} = Sd(\hat{u})$;
  if *convergence* **then**
    Return $\hat{\beta}_{LASSO}$ and report sucess
  else
    Go back to step 1
  end
  if *reweighted* **then**
    Set $\omega = \frac{1}{1/n+|\beta|}$ in which β is the parameters obtained in step 2.
  end
end

Algorithm 2: An algorithm for adaLASSO

3
Theory

In this chapter we will lay out the theory for two things:

1. That $\lambda = \sigma\sqrt{2\log(p)/n}$ allows us to say $\|X'u\|_\infty < \lambda$ with high probability
2. Under which conditions our algorithm converges to the true variance of the error

We will focus on model selection. It requires more hypothesis than if we focused on doing prediction. However, economists are usually interested in causal explanations and it requires knowing which variables are relevant and which are irrelevant. Another goal economists nowadays use variable selection is for choosing controls when estimating a treatment effect. For this goal, the conditions are milder than model selection.

There are numerous assumptions about the Data Generating Process (DGP) sufficient to prove the theorems bellow:

Assumption 1 *The true model is linear on the parameters:*

$$Y = X\beta^0 + u,$$

and X is independent of u

Assumption 2 *The true vector of coefficients β^0, can be sparse: $card(\beta_0) = s \leq p$*

Assumption 3 *The smallest eigenvalue of the sample covariance matrix of the active variables is bounded away from zero.*

Assumption 4 *u, the error vector, is independent and subgaussian with parameter σ*

Assumption 5 *X is deterministic*

Assumption 6 *Thw eights for the adaLASSO, ω_j, are $1/\sqrt{n} + |\hat{\beta}|$, in which $\hat{\beta}$ is a consistent estimator of the true vector of coefficients, β^0*

Hypothesis 1 allows the use of dictionary of variables, e.g. powers of variables. Hypothesis 2 allows for the possibility that the true vector of coefficients is sparse. We will always work with sparse coefficients in the simulations. Most bounds bellow depend on the cardinality of the true vector of coefficients, s. We can allow $s = p$, however it makes most of the bound bellow large and potentially useless if $p \to \infty$. Hypothesis 4 allows for gaussian errors with variance σ^2 and more general distributions that are not heavy tailed. Hypothesis 5 is a bit unusual in economics and Bühlmann & Van de Geer (2011) provides ways of relaxing it. Hypothesis 6 is similar to the hypothesis in Zou (2006). For models in which $p < n$, one could use the Least Squares estimator. For cases in which $p > n$, one could use the LASSO. Medeiros & Mendes (2016) provides guarantees that in a more general setting that ours, the weights coming from a first stage LASSO penalize more the coefficients that are zero than the non zero coefficients.

We will need also a hypothesis concerning both the data and the estimation process:

Assumption 7 *X satisfies the Restricted Eigenvalue (RE) condition with $(\kappa, 3)$, i.e.*

$$\frac{1}{n}\|X\Delta\|_2^2 > \kappa\|\Delta\|_2^2 \quad \forall\Delta \in \complement_\alpha(S),$$

in which $\complement_\alpha(S) = \{\Delta \in \mathbb{R}^p : \|\Delta_{S^C}\|_1 < \alpha\|\Delta_S\|_1\}$

Besides these hypothesis, we will also use the Basic Inequality, that comes from the basic optimality condition:

$$0 \leq \frac{\|X\hat{\Delta}\|_2^2}{n} \leq 2u'X\hat{\Delta}/n + 2\lambda(\|\beta^0\|_1 - \|\hat{\beta}\|_1), \tag{3-1}$$

where $\hat{\Delta} = \hat{\beta} - \beta_0$.

3.1 Convergence of the Algorithm

We want to show that the procedure outlined in Algorithm 1 converges. We ideally would like to show that it converges to somewhere near the real error variance. This is not always true for the LASSO.

On the other hand, the algorithm will always converge to some point. The reason for that is simple: the sequence of regularization parameters is monotonic and bounded, and any monotonic sequence that is bounded has a limit. For this consider the function:

$$\Lambda(\lambda) = \frac{\|Y - X\beta(\lambda)\|_2}{\sqrt{n}}\sqrt{2\log(p)/n}$$

Lets prove both claims. To see that it is bounded, notice that $\|\hat{u}\|_2$ is never larger than $\|y\|_2$. This will happen only if no variable is added to the model for a given λ. So starting the algorithm at $\|y\|_2/\sqrt{n}$ it can only go down: if, with $\sigma_k = \sigma_y$, no variable is added, then $\sigma_{k+1} = \sigma_y$ and the algorithm quits. On the other hand $\|\hat{u}\|_2$ is never smaller than zero, since it is a norm. A more statistical approach requires that we break this case in two: if there are more variables than observations, the LASSO will select a subset such that there are $n-1$ variables with coefficients different from zero. Since there is no penalization, we will fit the OLS estimate for that subset of variables. On the other hand, if $p < n$, then all variables will be on the model and we will have the OLS fit, which will generate $\|\hat{u}\|_2 \geq 0$

Now let's show that it is monotonic. To see that, assume that from the k to the $k+1$ interation we have $\lambda_{k+1} < \lambda_k$. This is equivalent to make the constraint less tight, and therefore $||\beta_{k+1}||_1 > ||\beta_k||_1$. Now notice that:

$$\min_{||\beta||_1 < R'} ||y - X\beta||_2^2 \leq \min_{||\beta||_1 < R} ||y - X\beta||_2^2 \tag{3-2}$$

If $R' > R$, since the solution of the problem on the right hand side is feasible for the left hand side. So, we have that $||y - X\beta_{k+1}||_2^2 \leq ||y - X\beta_k||_2^2$. This leaves two options: if $||y - X\beta_{k+1}||_2^2 = ||y - X\beta_k||_2^2$, the algorithm quits. If $||y - X\beta_{k+1}||_2^2 < ||y - X\beta_k||_2^2$, then $\lambda_{k+2} = ||u_{k+1}||_2/\sqrt{n}\sqrt{2\log(p)/n} < ||u_k||_2/\sqrt{n}\sqrt{2\log(p)/n} = \lambda_{k+1}$. If $\lambda_{k+1} > \lambda_k$, then we can apply the same argument to see that $\lambda_{k+2} \geq \lambda_{k+1}$.

Monotonicty and the fact that the algorithm only searches a limited space guarantees the existence of a fixed point, and the fact that iteration will reach a fixed point - this is guaranteed by the Tarski-Kantorovich Theorem (see the Appendix). The theorem also states that there will be a minimum and a maximal fixed point, and that in order to reach the minimum fixed point one needs that there exists a point such that $\lambda \geq \Lambda(\lambda)$; in order to reach the maximum fixed point, we need a point $\lambda \leq \Lambda(\lambda)$. We have both: if $\lambda = 0$, then $\Lambda(0) = \|Y - X\beta_{OLS}\|_2 A\sqrt{2\log(p)/n} \geq 0$, which can be equal to zero if $p \geq n$. Now, on the other side if we use $\lambda_{\sigma_y} = \sigma_y A\sqrt{2\log(p)/n}$, then $\forall\lambda > 0\ \|Y - X\beta(\lambda)\|_2/\sqrt{n} \leq \sigma_y$, and therefore $\Lambda(\sigma_y) \leq \sigma_y$.

Tarski Kantorovich Theorem does not tells us how many fixed point there are, or even what are their values. However, the existence of this fixed point and how to find it is also of independent interest: most theorem on the consistency of the LASSO depend on the fact that $\|2X'u/n\|_\infty < \lambda$. Frequently, people use the fact that $\sigma_u\sqrt{2\log(p)/n} > \|2X'u/n\|_\infty$ with high probability. Now, assume one uses a model estimated by the LASSO and selects the regularization parameter by any method. Then, if the standard deviation of the residual

implies that $\sigma\sqrt{2\log(p)/n} > \lambda$, the researcher faces an internal consistency problem: if his model is right, his choice of regularization parameter violates the most common bound given to guarantee the conditions for ℓ_2 consistency of the parameters.

Fortunately, we can get some bounds on the size of the error. Using Theorem 1, and after some algebra, equation iii yields:

$$\frac{||\hat{u} - u||_2}{\sqrt{n}} \leq 3\sqrt{\frac{s}{\kappa}}\lambda$$

Denote $\hat{u}_k$ the residual obtained by the k step of Algorithm 1. Then, substituting our choice of λ, we get:

$$\frac{||\hat{u}_k - u||_2}{\sqrt{n}} \leq 3A\frac{||\hat{u}_{k-1}||_2}{n}\sqrt{\frac{2s\log(p)}{\kappa}}$$

Cancel the $1/\sqrt{n}$ on both sides to get:

$$||\hat{u}_k - u||_2 \leq 3A\frac{||\hat{u}_{k-1}||_2}{\sqrt{n}}\sqrt{\frac{2s\log(p)}{\kappa}} \tag{3-3}$$

Since we do not have the distance between the previous estimation and the true error in the right hand side, we cannot use stronger results to characterize the fixed point. One could be tempted to pick A in such a way that this bound is really small. However, in all theorems above we made the hypothesis that $\lambda > \|2u'X/n\|_\infty$. Choosing an A too small will lead to a violating of this hypothesis. In the next section we will show some results that allow us to get around this.

We will also work with the adaLASSO, and while the proofs of the LASSO can be carried for the adaLASSO case, we can prove conditions for the adaLASSO that allow more control over the bounds. Let's start with the weighted LASSO:

$$\min_\beta \|y - X\beta\|_2^2 + \lambda\|W\beta\|_1,$$

in which W is a $p \times p$ diagonal matrix of weights. We can rewrite the expression above by setting $\beta_w = W\beta$ and we will get:

$$\min_\beta \|y - XW^{-1}\beta_w\|_2^2 + \lambda\|\beta_w\|_1$$

We will define $\|\beta\|_{w1} := \|W\beta\|_1$, the weighted ℓ_1 norm. We can have a basic inequality for this new penalty that is:

$$0 < \frac{1}{n}\|X\hat{\Delta}\|_2^2 < \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\beta^0\|_{w1} - \|\hat{\beta}\|_{w1}),$$

and $\hat{\Delta} = \beta^0 - \hat{\beta}$. We can also define a $\mathbb{C}_\alpha(S)$ cone with respect to $\|.\|_{w1}$:

$$\mathbb{C}_\alpha^{w1}(S) = \{\Delta \in \mathbb{R}^p \, \|\Delta_{S^c}\|_{w1} < \alpha \|\Delta_S\|_{w1}\},$$

and we can define a RE condition with respect to this new cone, with parameters (κ, α), which we will call the weighted RE condition:

$$\kappa_w \|\hat{\Delta}_w\|_2^2 \leq \frac{1}{n} \|XW^{-1}\hat{\Delta}_w\|_2^2$$

We will work with assumption 1 unaltered and A2A:

Assumption 2A *X satisfies the weighted RE condition with $(\kappa, 3)$*

Now we can do a slight change to Theorem A of Appendix I to use the Weighted Eigenvalue condition:

Theorem 3.1 *With $\lambda \geq \|\frac{2}{n} u' X W^{-1}\|_\infty$ and the Weighted RE condition $(\kappa, 3)$:*

$$\|\hat{\beta} - \beta^0\|_2 \leq \frac{3}{\kappa_w} \lambda \sqrt{s}$$

This might not seem like a big change from the LASSO to the adaLASSO, however notice that the weighted RE condition allows us to write, for the case in which the Gram Matrix is the identity:

$$\begin{aligned} \kappa_w \|\hat{\Delta}_w\|_2^2 &\leq \frac{1}{n} \hat{\Delta}_w' W^{-1} X' X W^{-1} \hat{\Delta}_w \\ \kappa_w \|\hat{\Delta}_w\|_2^2 &\leq \hat{\Delta}_w' W^{-1} W^{-1} \hat{\Delta}_w \\ \kappa_w \|\hat{\Delta}_w\|_2^2 &\leq \hat{\Delta}_w' W^{-2} \hat{\Delta}_w \end{aligned} \tag{3-4}$$

Now, since W is just a diagonal matrix that can be written as a vector ω with size p and so $\hat{\Delta}_w' W^{-2} \hat{\Delta}_w = \sum_{j=1}^p \omega_j^{-2} \hat{\Delta}_{wj}^2$ and so:

$$\kappa_w \|\hat{\Delta}_w\|_2^2 \leq \sum_{j=1}^p \omega_j^{-2} \hat{\Delta}_{wj}^2 \leq \max_{j=1,\ldots,p} \omega_j^{-2} \|\hat{\Delta}_w\|_2^2$$

So $\kappa_w \leq \max_{j=1,\ldots,p} \omega_j^{-2}$, which is possibly a really large number and helps in our contraction argument. Notice that in the case of a identity Gram matrix, $\kappa = 1$.

This result is true for any set of weights. It does not mean that it is always useful, since a random set of weights might not generate a useful inequality. One could also argue that we could choose the weights such that $\|\omega^{-2}\|_2$ was as large as possible. To avoid this complications, we work with the weights that are the inverse of the absolute value of the LASSO.

3.2 Regularization Parameter

So what about our regularization parameter? We require that $\lambda > \|2u'X/n\|_\infty$. While the proof for the case $\lambda = 2\sigma\sqrt{2\log(p)/n}$ is available in Bickel et al. (2009), Bühlmann & Van de Geer (2011) and Wainwright (2019), we will show what happens when $\lambda = A\sigma\sqrt{2\log(p)/n}$.

Theorem 4 justifies why $A > 2$ in Bickel et al. (2009). Otherwise, $2p^{1-A^2/4}$ would diverge and $\lambda > \|2u'X/n\|_\infty$ would not happen with high probability. This means that our proposal of $A = 1$ would not work for the LASSO.

The adaLASSO, on the other hand, requires a different event: $\lambda > \|2u'XW^{-1}\|_\infty$. This gives us a lot more room:

Theorem 3.2 *Assume X fixed and u be subgaussian with parameter σ and let $\omega_j = (1/\sqrt{n} + |\beta_L|)^{-1}$, in which β_L is some consistent estimator of the coefficients. Then $P(\sigma A\sqrt{2\log(p)/n} > \|2u'XW^{-1}/n\|_\infty) > 1 - 2p^{1-(\omega_j \min A)^2/4}$, in which $\omega_{\min}$ is the smallest weight.*

Proof. See the Appendix ■

4
Simulations

This chapter shows Monte Carlo experiments. We have two sets of experiments: the first one investigate the convergence of the algorithm. We show that the algorithm using adaLASSO has better convergence properties than the one using the LASSO.

The second set of experiments shows how the algorithm behave, with respect to model selection and forecasting. We compare it with some alternatives and show that for a number of cases, particularly when we have many variables in the active set, it behaves reasonable well.

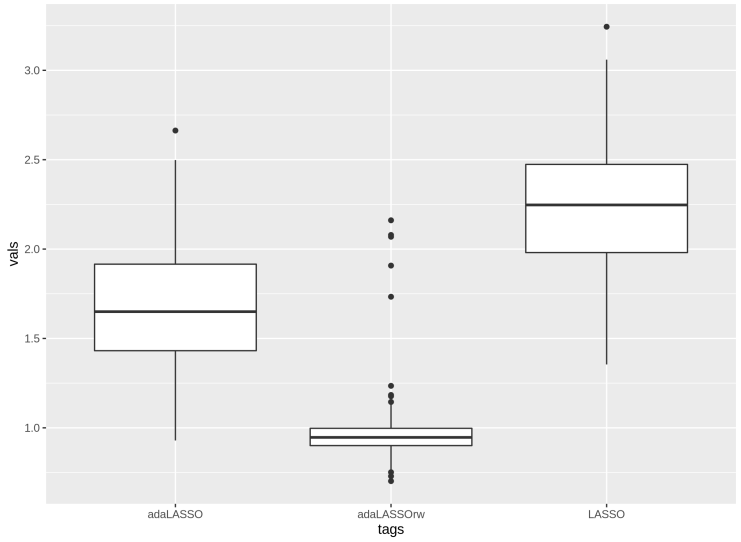
4.1
Convergence

We begin analysing when the algorithm converges. Notice that since the only part of λ that is updated is the standard deviation of the error, we will analyse the convergence of the standard deviation of the error. In all simulations of this subsection, we have X i.i.d. from a standard normal and the error also comes from a standard normal. Using an i.i.d design with covariance matrix equals to the identity allows us to say that $\kappa = 1$. In this section, we will always make 2000 replications for each experiment. We will vary the sample size (n), the number of variables included (p) and the size of the active set (s). We conjecture that convergence of the algorithm depends on $3\sigma\sqrt{2s\log(p)/n}$ and controlling for the three parameters above allow us to control $3\sigma\sqrt{2s\log(p)/n}$. Smaller values for it should give better convergence and simulations back this claim.

In our first experiment, we set $n = 100$, $p = 50$ and $s = 10$. These values imply $3\sigma\sqrt{2s\log(p)/n} = 2.65$. Our initial guess for the variance of the error is the standard deviation of the dependent variable, which is around $\sqrt{11}$. We show the results in Figure 4.1. We show the results for the LASSO and two cases of the adaLASSO: updating the weights at each iteration and not updating the weight at each iteration. The former corresponds to adaLASSOrw in the figure.

It is clear that the LASSO does not converge to the true value of the standard deviation of the error. The adaLASSO in which the weights are not

Figure 4.1: Experiment 1. Boxplot of estimated variances



updated makes things a lot better. On the other hand, always re-estimating the weights allow to the adaLASSO to get the standard deviation with much more precision.

It also shows a nice feature of the algorithm: in no case the standard deviation of the error diverges. As a matter of fact, we never reach the limit of iterations. This backs the claim that the algorithm will always converge, but not always to the right point.

The second experiment keeps all the parameters above the same, but change the initial guess of the standard deviation to 0.5. A better guess and a finite number of iterations should cause a better estimation of the standard deviation of the error by the LASSO and the adaLASSO not re-weighted, as we show in Figure 4.2. The gains for the adaLASSO not re weighted are clear, while the gains for the LASSO are less clear.

Experiment three only changes the sample size, to $n = 1000$, so $3\sigma\sqrt{2s\log(p)/n} = 0.83$. Figure 4.3 shows the estimates for this case. Notice the change of the axis y: the LASSO comes down from almost double the true value of the standard deviation of the error, in the case of experiments above, to a value 5% above the true value - and the “outliers" are a bit above 10% off.

Experiment four changes the sample size to 400 and $3\sigma\sqrt{2s\log(p)/n} = 1.32$. The results are in Figure 4.4. Again, the LASSO does not converge and the adaLASSO shows better properties. Experiment five is closely related, and uses $n = 400$ and $s = 5$ and therefore $3\sigma\sqrt{2s\log(p)/n} = 0.93$. The objective is to show that is not only sample size that matters, but actually all the three elements: the size of the active set, the number of variables included and the

Figure 4.2: Experiment 2. Boxplot of estimated variances

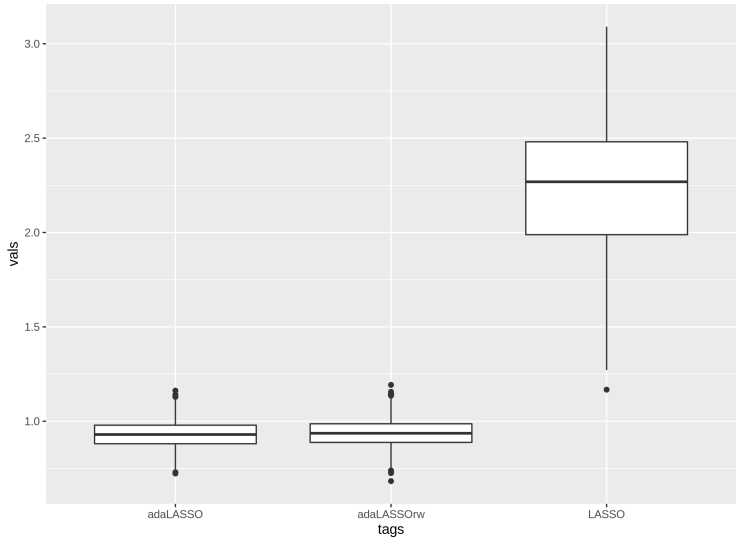
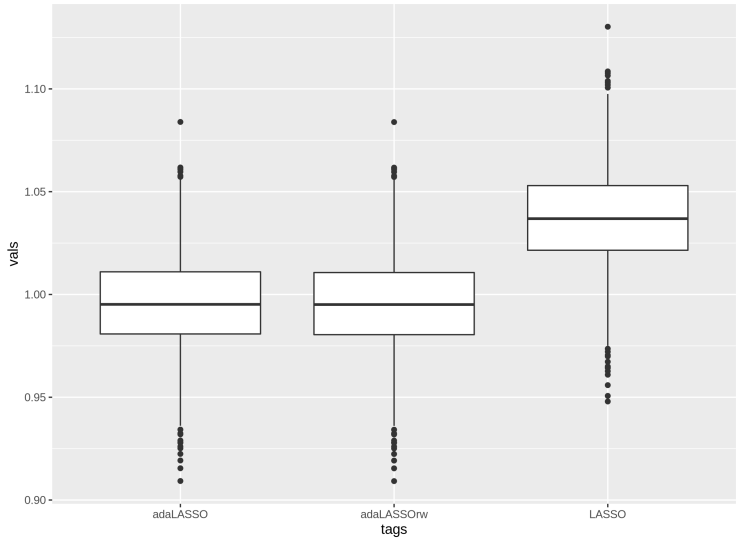


Figure 4.3: Experiment 3. Boxplot of estimated variances



sample size. The results are illustrated in Figure 4.5.

4.2 Model selection

Building on Coutinho et al. (2017), we will test the our method (NM) for selecting the regularization parameter against three competitors: adaLASSO using Cross Validation (CV) and the BIC (Bayesian Information Criteria) for the regularization parameter and the hdm package[1], that is based on Belloni et al. (2013). We always let the regularization parameter change from the first step estimation for the second step estimation (unlike Coutinho et al. (2017)).

[1]We change the default to let the package assume that the error is homoscedastic. The results are even worse when we use the default

Figure 4.4: Experiment 4. Boxplot of estimated variances

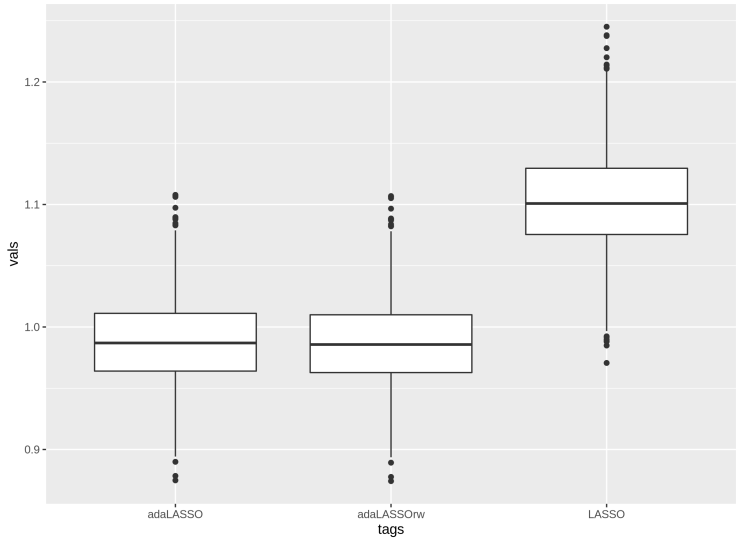
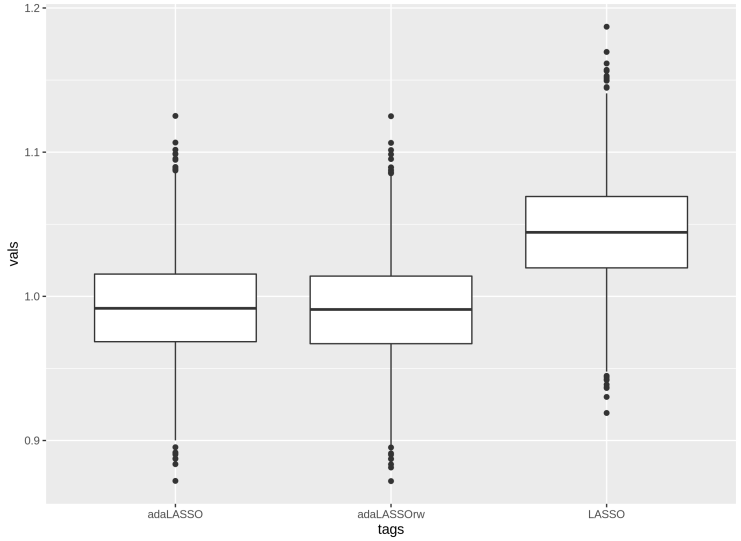


Figure 4.5: Experiment 5. Boxplot of estimated variances



We have six designs, and the results are reported in Tables 4.1 to 4.6 . The variable *Non zeros Right* pick how many relevant coefficients were recovered; the *Zeros Right* pick how many irrelevant coefficients were set to zero; the *Right model* is a dummy that is equals to 1 if in a given simulation the method been tested recovered *all* the variables, setting the irrelevant variables to zero and the relevant variables different of zero. The design of the Monte Carlo simulations are:

1. $n = 100, \sigma^2 = 3$, 20 relevant variables and 30 irrelevant variables.

2. $n = 100, \sigma^2 = 3$, 10 relevant variables and 40 irrelevant variables.

3. $n = 100, \sigma^2 = 1$, 10 relevant variables and 40 irrelevant variables.

4. $n = 1000, \sigma^2 = 1$, 40 relevant variables and 60 irrelevant variables.

5. $n = 100, \sigma^2 = 1$, 10 relevant variables and 90 irrelevant variables.

6. $n = 100, \sigma^2 = 1$, 30 relevant variables and 70 irrelevant variables.

We set that if the variable j is relevant, then $\beta_j = 1$. Otherwise, $\beta_j = 0$. Designs 1 and 6 are particularly tricky for the LASSO due to the high variance and the fact that the model is not as sparse as the others, respectively. Only design 5 and 6 can be considered “big data" i.e. $p \geq n$.

Table 4.1: Design 1: Result for 5000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	1.00	0.85	0.04
CV	0.99	0.90	0.10
NM	0.97	0.98	0.44
HDM	0.27	0.99	0.00

Table 4.2: Design 2: Result for 3000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	1.00	0.93	0.17
CV	0.99	0.96	0.35
NM	1.00	0.97	0.31
HDM	0.73	0.99	0.14

Table 4.3: Design 3: Result for 1000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	1.0000	0.9647	0.3940
CV	1.0000	0.9830	0.6990
NM	1.0000	0.9968	0.8780
HDM	0.9926	0.9811	0.4650

Table 4.4: Design 4: Result for 1000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	1.0000	0.9976	0.9090
CV	1.0000	0.9999	0.9920
NM	1.0000	0.9997	0.9800
HDM	1.0000	0.9987	0.9250

The results show that our option is not always the best: for designs 2 and 4, the CV is better than our method, but not by much. On the other hand, for designs 1 and 6, our method dominates the other options and is better in the remaining tests.

Table 4.5: Design 5: Result for 1000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	0.995	0.280	0.033
CV	1.000	0.983	0.525
NM	1.000	0.996	0.723
HDM	0.984	0.983	0.241

Table 4.6: Design 6: Result for 1000 replications

	Non Zeros Right	Zeros Right	Right model?
BIC	1.00	0.93	0.04
CV	1.00	0.94	0.03
NM	0.90	0.98	0.32
HDM	0.09	1.00	0.00

It's interesting to note that designs 1 and 6 show clearly the trade off between getting more zeros right and getting the non zeros right: our method is worst than CV if our main concern is to include all the relevant regressors and is worst than the `HDM` in exclude the irrelevant variables. However, by allowing the possibility of exclusion of more variables than CV and less than the HDM, we are able to set more right zeros. Its interesting to note that this actually makes our method better than the others in situations in which the model is *less* sparse, namely designs 1 and 6.

Table 4.7, we repeat the same designs as above. However, instead working with a random X and a random u, we keep X fixed. This is more in line with the hypothesis we used in the theory. This should make easier to recover the right model, and the simulations back it, although the difference is not dramatic.

Table 4.7: Simulations with fixed design

	Non Zeros Right	Zeros Right	Model Right
Design 1	0.976	0.985	0.473
Design 2	0.998	0.970	0.290
Design 3	1.000	0.992	0.714
Design 4	1.000	0.999	0.967
Design 5	1.000	0.997	0.727
Design 6	1.000	0.996	0.743

Table 4.8 shows the simulations using an error with chi-squared distribution - a distribution that is not subgaussian, but is subexponential. We change the number of degrees of freedom in order to change the variance. We keep X fixed between simulations and make 2000 replications. The number of observations and variables and relevant variables are the same as the designs above. The performance is worse than the case with gaussian errors, which is

unsurprising since the theory is based on the hypothesis that the errors are subgaussian.

Table 4.8: Simulations with Subexponential error

	Non Zeros Right	Zeros Right	Model Right
Design 1	0.984	0.985	0.577
Design 2	0.997	0.971	0.314
Design 3	1.000	0.992	0.733
Design 4	1.000	0.999	0.959
Design 5	1.000	0.991	0.463
Design 6	0.984	0.992	0.636

Tables 4.9, 4.10 and 4.11 show the results of simulations using the algorithm when we use Student's t distribution for the error with different degrees of freedom, with 3000 replications each. We use a fixed design for X and the designs are the same as in the previous simulations. We drop design 2 since the only change between it and design 1 is the variance of the error. Notice that these designs are not completely equivalent to the previous designs, since setting the degrees of freedom define the variance of the distribution. The number of observations and variables and relevant variables are the same as the designs above. The fact that the variances are not the same make it harder to compare these results with the previous results. However, more degrees of freedom make the distributions more well behaved and we would expect better results as the degrees of freedom increase. The model right column illustrates exactly that.

Table 4.9: Simulations with Polynomial Tails: Student's t Distribution with 4 degrees of freedom

	Non Zeros Right	Zeros Right	Model Right
Design 1	0.99	0.99	0.72
Design 3	1.00	0.98	0.45
Design 4	1.00	1.00	0.85
Design 5	0.99	0.99	0.32
Design 6	0.97	0.98	0.35

4.3 Regularization Parameter

Using $\lambda = \sigma\sqrt{2\log(p)/n}$ instead of using $\lambda = 2\sigma\sqrt{2\log(p)/n}$ is another innovation that needs backing. In this section, we show some Monte Carlo simulation that compares this to the original proposal made in Bickel et al. (2009). We gave an explanation on why it is not problematic when used with the adaLASSO, and therefore we will compare all results using the adaLASSO.

Table 4.10: Simulations with Polynomial Tails: Student's t Distribution with 8 degrees of freedom

	Non Zeros Right	Zeros Right	Model Right
Design 1	1.00	1.00	0.90
Design 3	1.00	0.99	0.61
Design 4	1.00	1.00	0.93
Design 5	1.00	0.99	0.55
Design 6	0.98	0.99	0.50

Table 4.11: Simulations with Polynomial Tails: Student's t Distribution with 3 degrees of freedom

	Non Zeros Right	Zeros Right	Model Right
Design 1	0.97	0.98	0.59
Design 3	0.99	0.97	0.36
Design 4	1.00	1.00	0.75
Design 5	0.99	0.98	0.18
Design 6	0.96	0.97	0.23

We start with design 1, and the results are showed on Table 4.3. Both are fitted using the same algorithm, all that changes is the value of A.

Table 4.12: Design 1, 2000 replications

	Non Zero Right	Zero Right	Model right
$A = 1$	0.97	0.98	0.45
$A = 2$	0.59	0.99	0.00

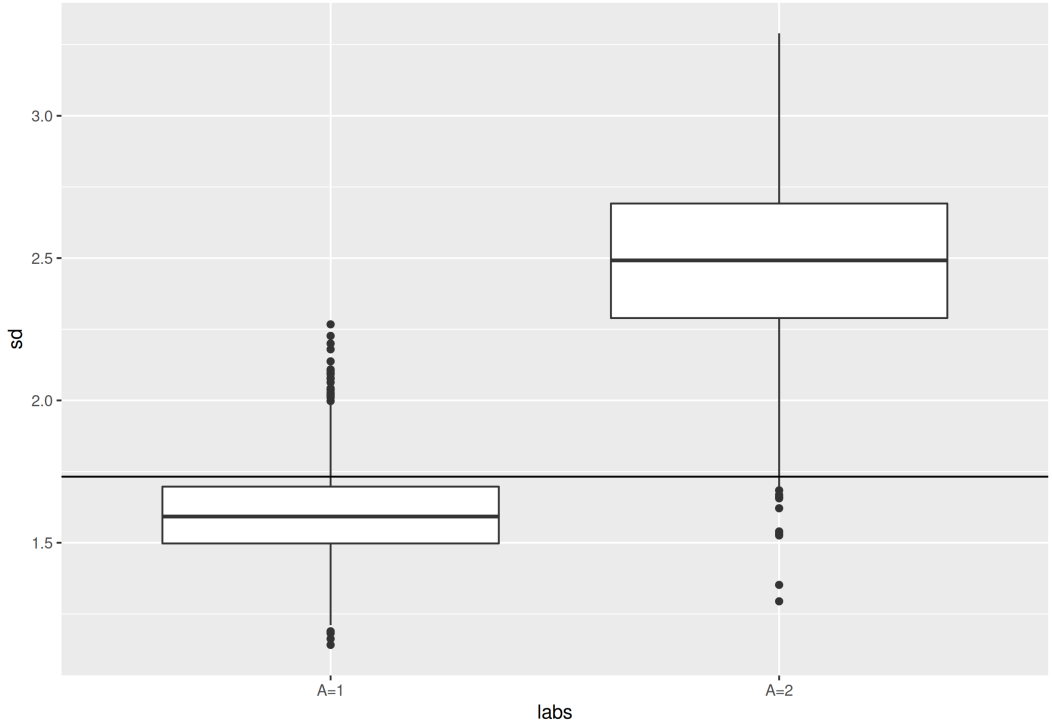
Looking at the model right column, using $A = 1$ is better then $A = 2$ for this design. All the gain comes from the fact that we get more non zero coefficients right then using $A = 2$. In other words, in a design in which we have a lot of noise and a lot of relevant variables, the regularization parameter from Bickel et al. (2009) do not let enough coefficients be different then zero. Notice that both benefit from using adaLASSO, but since $A = 1$ was engineered to work with adaLASSO, it works better than the alternative.

It could be that in cases in which the variance of the error is smaller and we have less relevant variables we perform (much) worse. We then test design 3, which is shown in Table 4.3. Notice that using $A = 2$ still beats BIC and HDM from Table 4.3.

Table 4.13: Design 3, 2000 replications

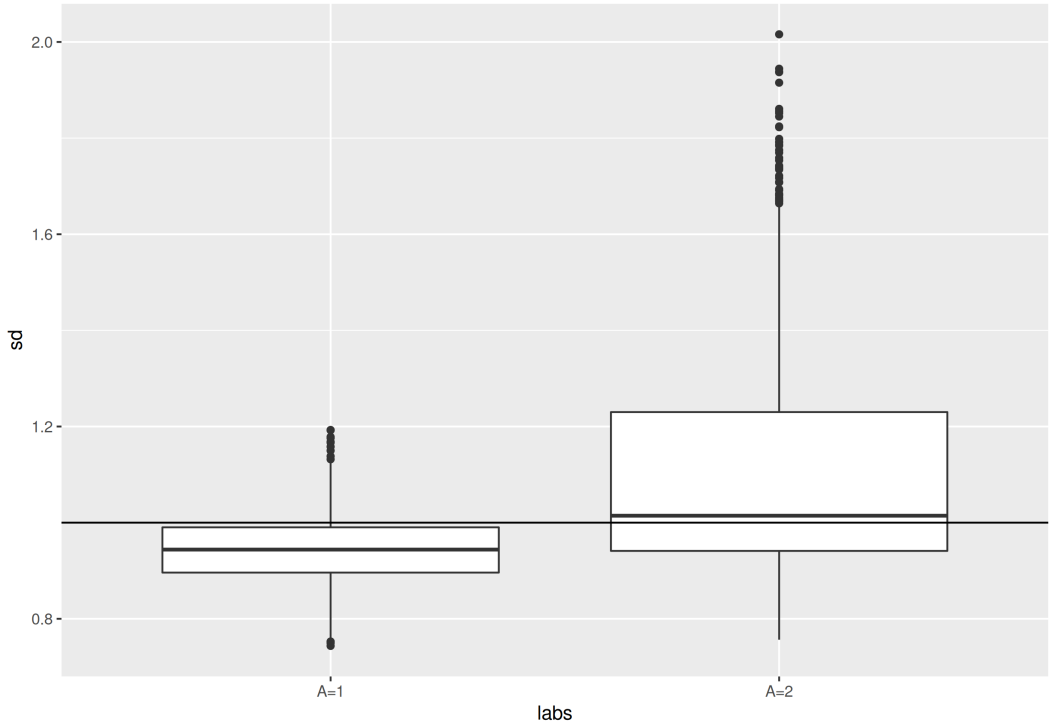
	Non Zero Right	Zero Right	Model right
$A = 1$	1.0000	0.9966	0.8710
$A = 2$	0.9445	0.9997	0.6670

Figure 4.6: Boxplot of estimated standard deviation



A valid worry is whatever we have a situation in which we have two errrors cancelling each other. Our method could be overestimating the variance in both cases, and $A = 1$ is just correcting for the bias. Figures 4.6 and 4.7 show the estimated residual standard deviation for Tables 4.3 and 4.3, respectively. The horizontal line mark the true standard deviation. Notice that for Design 3, using $A = 2$ generates gross errors. Even for Design 1, the residual standard deviation is a lot more spread than for $A = 1$.

Figure 4.7: Boxplot of estimated standard deviation



5
Empirical Example

As an empirical example, we repeat the regressions made by Donohue & Levitt (2001) over the effects of abortion over crimes. We follow closely the replication done by Belloni et al. (2013), using their data and their program to generate regressors to be selected via adaLASSO. We will compare our implementation with two different guesses to the variance and compare the results to the `HDM` package in R[1]. The initial guess is 1 and the lower guess is 0.1. Results for the coefficients and standard errors are reported in Tables 5.1, 5.2 and 5.3.

Table 5.1: Coefficients: Effect over Violent Crimes

	Coef	SE	t-Stat	P-value
HDM	-0.17	0.12	-1.41	0.16
Us	-0.28	0.13	-2.17	0.03
Us - Lower Guess	-0.09	0.13	-0.63	0.53

Table 5.2: Coefficients: Effect over Property Crimes

	Coef	SE	t-Stat	P-value
HDM	-0.12	0.42	-0.28	0.78
Us	-0.05	0.05	-1.04	0.30
Us - Lower Guess	-0.10	0.63	-0.16	0.87

Table 5.3: Coefficients: Effect over Murder

	Coef	SE	t-Stat	P-value
HDM	-0.12	0.42	-0.28	0.78
Us	-0.11	0.45	-0.24	0.81
Us - Lower Guess	-0.10	0.63	-0.16	0.87

There are differences between our algorithm with different guesses. However, considering the amount of regressors and the size of the series, previous results say that the algorithm won't necessarily converge to a single value, which explains the difference between the guesses. The results are

[1]Even the Matlab programs available online do not replicate the results they report in Belloni et al. (2013)

more scattered for the coefficient over violent crimes, with one estimate being significant, while the estimates are incredibly concentrated for murders.

To understand better what each algorithm is doing, Tables 5.4, 5.5 and 5.6 show how many regressors are select by each method in both the Outcome and Treatment regressions

Table 5.4: Number of variables selected: Violent Crimes

	HDM	Us	Us - Lower Guess
Outcome $\sim x$	3	2	40
Treat $\sim x$	12	9	26

Table 5.5: Number of variables selected: Property Crimes

	HDM	Us	Us - Lower Guess
Outcome $\sim x$	6	2	32
Treat $\sim x$	14	11	30

Table 5.6: Number of variables selected: Murder

	HDM	Us	Us - Lower Guess
Outcome $\sim x$	0	2	72
Treat $\sim x$	9	10	22

In line with the results from the simulation, our algorithm is able to select more variables than `HDM`. The effect is larger when we lower the starting guess of the variance. This might explain the difference in the coefficients we obtain.

6
Conclusion

This paper presents yet another way to select the regularization parameter. We use both the theory and the data to select the regularization parameter. In the end, we have a relatively simple algorithm that is useful - as shown by the simulations.

Using the adaptive LASSO instead of the LASSO proves to be important for the convergence of the algorithm. The adaptive LASSO also plays a key role for variable selection - this was the main point of Zou (2006). Our simulations point to the potential of the adaLASSO, especially in challenging problems that are not “too sparse". However, its non asymptotic theory is not completely developed.

The simulation results are encouraging about the effectiveness of the algorithm. However, it still requires that the user sets a initial guess for the variance, and the result can be quite sensitive to the initial guess. Understanding the sensitivity and what is the optimal initial guess - if there is one - would be an important direction to make the algorithm easier to use.

Bibliography

TIBSHIRANI, R.. **Regression Selection and Shrinkage via the Lasso**, 1996.

BICKEL, P. J.; RITOV, Y. ; TSYBAKOV, A. B.. **Simultaneous analysis of lasso and dantzig selector**. Ann. Stat., 2009.

BÜHLMANN, P.; VAN DE GEER, S.. **Statistics for High-Dimensional Data: Methods, Theory and Applications**. Springer, 2011.

WAINWRIGHT, M. J.. **High-dimensional statistics: A non-asymptotic viewpoint**, volumen 48. Cambridge University Press, 2019.

ZOU, H.. **The adaptive lasso and its oracle properties**. Journal of the American Statistical Association, 2006.

ZHANG, Y.; LI, R. ; TSAI, C. L.. **Regularization parameter selections via generalized information criterion**. J. Am. Stat. Assoc., 2010.

FAN, Y.; TANG, C. Y.. **Tuning parameter selection in high dimensional penalized likelihood**. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 2013.

HUI, F. K.; WARTON, D. I. ; FOSTER, S. D.. **Tuning Parameter Selection for the Adaptive Lasso Using ERIC**. Journal of the American Statistical Association, 2015.

BELLONI, A.; CHEN, D.; CHERNOZHUKOV, V. ; HANSEN, C.. **Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain**. Econometrica, 80(6):2369–2429, 2012.

BELLONI, A.; CHERNOZHUKOV, V. ; HANSEN, C.. **Inference on treatment effects after selection among high-dimensional controls**. Review of Economic Studies, 2013.

COUTINHO, D.; MEDEIROS, M. ; SOUZA, P.. **The Illusion of Independence: High Dimensional Data, Shrinkage Methods and Model Selection**. 2017.

HASTIE, T.; TIBSHIRANI, R. ; FRIEDMAN, J.. **Elements of Statistical Learning 2nd ed.** 2009.

MEDEIROS, M.; MENDES, E.. **L1-regularization of high-dimensional time-series models with non-gaussian and heteroskedastic errors**. Journal of Econometrics, 191, 11 2016.

DONOHUE, J. J.; LEVITT, S. D.. **The impact of legalized abortion on crime**. Quarterly Journal of Economics, 2001.

GRANAS, A.; DUGUNDJI, J.. **Fixed Point Theory**. Springer Monographs in Mathematics. Springer New York, 2013.

COLEMAN, W. J.. **Equilibrium in a production economy with an income tax**. Econometrica, 59(4):1091–1104, 1991.

Appendix I: Theorems

Theorem 1 *Under Assumptions 1 and 2 and* $\lambda > max_{j=1,..J}|2u'X_j|/n$*, we have:*

$$\|\hat{\beta} - \beta^0\|_2 < \frac{3}{\kappa}\sqrt{s}\lambda \tag{ii}$$

$$\frac{\|X(\beta^0 - \hat{\beta})\|_2^2}{n} \leq \frac{9s\lambda^2}{\kappa} \tag{iii}$$

The proof can be found in Wainwright (2019)

Theorem 2 (Banach Fixed Point Theorem) *Let* $f : \mathcal{A} \to \mathcal{A}$ *and* $x, y \in \mathcal{A}$ *and* $(\mathcal{A}, d)$ *is a complete metric space. If:*

$$d(f(x), f(y)) \leq hd(x, y)$$

For $h < 1$*, then* f *has a unique fixed point that is reached from any point by the sequence* $x_0 \in \mathcal{A}, x_n = f^n(x_0)$ *and* f *is called a* contraction map

Theorem 3 (Tarski Kantorovich Fixed Point) *Let* $(P, \preceq)$ *be a partially ordered set and* $F : P \to P$ *continuous. Assume that* $b \in P$ *and:*

- *if* $x \succeq y$*,* $F(x) \succeq F(y)$
- $b \preceq F(b)$
- *Every countable chain in* $\{x|x \succeq b\}$ *has a supremum*

Then F *has a fixed point* $\mu = \sup_n F^n(b)$ *and* μ *is the infimum of the set of fixed points of* F *in* $\{x|x \succeq b\}$

For a proof, see Granas & Dugundji (2013). For an application similar to the one we do here, see Coleman (1991).

Theorem 4 *Assume* X *fixed and* u *be subgaussian with parameter* σ*. Then* $P(\sigma A\sqrt{2\log(p)/n} > \|2u'X/n\|_\infty) > 1 - 2p^{1-A^2/4}$

For a proof of this Theorem, se Bickel et al. (2009)

Appendix II: Proofs

Theorem 4

We will show that by using the concentration bound for $\sigma A\sqrt{2\log(p)/n} < \|2u'XW^{-1}/n\|_\infty$, the complementary event. The event $\|2u'XW^{-1}\|_\infty > \lambda$ is equal to $\cap_{j=1}^{p}|2u'X_j\omega_j^{-1}| > \lambda$ and using DeMorgan's law we get the complementary event is $\cup_{j=1}^{p}|2u'X_jW_{jj}^{-1}| \geq \lambda$ and plug in our regularization parameter:

$$P\left(\bigcup_{j=1}^{p}|2u'X_j\omega_j^{-1}| \geq A\sigma\sqrt{2\log(p)/n}\right)$$

Boole's law gives that:

$$P\left(\bigcup_{j=1}^{p}|2u'X_j\omega_j^{-1}|/n \geq A\sigma\sqrt{2\log(p)/n}\right) \leq \sum_{j=1}^{p}P\left(|2u'X_j\omega_j^{-1}|/n \geq A\sigma\sqrt{2\log(p)/n}\right) \tag{iv}$$

Use again the facts that u is subgaussian and we have a fixed design. Also, notice that $\omega_j^{-1} = 1/\sqrt{n}+|\beta_L|$ .Apply Chernoff Bounds to the probability above:

$$\sum_{j=1}^{p}P\left(2u'X_j\omega_j^{-1}/n \geq A\sigma\sqrt{2\log(p)/n}\right) \leq \sum_{j=1}^{p}2\exp\left(-\frac{2A^2\sigma^2\log(p)/n}{2\times 4\omega_j^{-2}\sigma^2/n}\right) \leq 2p^{1-(A\omega_{\min})^2/4}$$

Since $\omega_{\min} \leq \omega_j \therefore -\omega_{\min} \geq -\omega_j \forall j = 1,\ldots,p$ ■

Lemma 1

Lemma 1 *The weighted LASSO solution belongs to* $\mathbb{C}_3^{w1}$ *for* $\lambda \geq \|2u'XW^{-1}/n\|_\infty$

Proof. Start with the basic inequality:

$$0 < \frac{1}{n}||X\hat{\Delta}||_2^2 < \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\beta^0\|_{w1} - \|\hat{\beta}\|_{w1})$$

Now, $\hat{\beta} = \beta^0 - \hat{\Delta}$, and substituting this on the norm we get:

$$\|\hat{\beta}\|_{w1} = \|\beta^0 - \hat{\Delta}\|_{w1} = \|\beta^0_S - \hat{\Delta}_S\|_{w1} + \|\hat{\Delta}_{S^c}\|_{w1}$$

Plug the expression above on the basic inequality:

$$0 < \frac{1}{n}||X\hat{\Delta}||_2^2 < \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\beta^0\|_{w1} - (\|\beta^0_S - \hat{\Delta}_S\|_{w1} + \|\hat{\Delta}_{S^c}\|_{w1}))$$

Use the inverse triangle inequality on $\|\beta^0_S - \hat{\Delta}\|_{w1}$:

$$\|\beta^0 - \hat{\Delta}\|_{w1} \geq \left|\|\beta^0\|_{w1} - \|\hat{\Delta}\|_{w1}\right|$$

This allows us to rewrite the basic inequality:

$$0 < \frac{1}{n}||X\hat{\Delta}||_2^2 \leq \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\beta^0\|_{w1} - (\|\beta^0_S\|_{w1} - \|\hat{\Delta}_S\|_{w1} + \|\hat{\Delta}_{S^c}\|_{w1})) = \\ = \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\hat{\Delta}_S\|_{w1} - \|\hat{\Delta}_{S^c}\|_{w1})$$

Rewrite $2/nu'X\hat{\Delta}$ as $2/nu'XW^{-1}W\hat{\Delta}$ and use Hölder Inequality to get:

$$\frac{2}{n}u'X\hat{\Delta} \leq |\frac{2}{n}u'X\hat{\Delta}| \leq \|\frac{2}{n}u'XW^{-1}\|_\infty \|W\hat{\Delta}\|_1 = \|\frac{2}{n}u'XW^{-1}\|_\infty \|\hat{\Delta}\|_{w1}$$

The last equality comes from the definition of $\|.\|_{w1}$. Plug it once again in the basic inequality:

$$0 < \frac{1}{n}||X\hat{\Delta}||_2^2 \leq \frac{2}{n}u'X\hat{\Delta} + 2\lambda(\|\hat{\Delta}_S\|_{w1} - \|\hat{\Delta}_{S^c}\|_{w1}) \leq \|\frac{2}{n}u'W^{-1}X\|_\infty \|\hat{\Delta}\|_{w1} + 2\lambda(\|\hat{\Delta}_S\|_{w1} - \|\hat{\Delta}_{S^c}\|_{w1})$$

We swapped W^{-1} because it is a diagonal matrix. Use that $\lambda > \|2/nu'W^{-1}X\|_\infty$ to get:

$$0 < \lambda(\|\hat{\Delta}\|_{w1} + 2\|\hat{\Delta}_S\| - 2\|\hat{\Delta}_{S^c}\|_{w1}) = \lambda(\|\hat{\Delta}_S\|_{w1} + \|\hat{\Delta}_{S^c}\|_{w1} + 2\|\hat{\Delta}_S\| - 2\|\hat{\Delta}_{S^c}\|_{w1}) = \\ = \lambda(3\|\hat{\Delta}_S\|_{w1} - \|\hat{\Delta}S^c\|_{w1})$$

■