UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 13-14
Statistics and Econometrics Series 13
May 2013

Departamento de Estadística
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34) 91 624-98-49

Lasso Variable Selection in Functional Regression

Nicola Mingotti, Rosa E. Lillo, Juan Romo.

Abstract

Functional Regression has been an active subject of research in the last two decades but still lacks a secure variable selection methodology. Lasso is a well known effective technique for parameters shrinkage and variable selection in regression problems. In this work we generalize the Lasso technique to select variables in the functional regression framework and show it performs well. In particular, we focus on the case of functional regression with scalar regressors and functional response. Reduce the associated functional optimization problem to a convex optimization on scalars. Find its solutions and stress their interpretability. We apply the technique to simulated data sets as well as to a new real data set: car velocity functions in low speed car accidents, a frequent cause of whiplash injuries. By "Functional Lasso" we discover which car characteristics influence more car speed and which can be considered not relevant.

Keywords: Norm one penalization; Variable selection; Algebraic re- duction; Convex optimization; Computer algebra.

*Nicola Mingotti is Phd student in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: nicola.mingotti@uc3m.es). Rosa Lillo is Professor in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: rosaelvira.lillo@uc3m.es). Juan Romo is Professor in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: juan.romo@uc3m.es). This research was supported in part by Spanish Ministry of Education and Science grants MEC 2009/00035/001, ECO2011-25706 and SEJ2007. Our investigations about whiplash injury in car accidents are made in collaboration with AXA, Madrid. In the specific case of this article, the company provided an important support in mining out of the market car characteristics that would have been difficult to obtain by other means. We want also to acknowledge Centro Zaragoza for support in understanding more about car accidents and finally, we acknowledge the Arbeitsgruppe fur Unfallmechanik for providing their car accidents database and for assistance in our first steps of deployment.

Lasso Variable Selection in Functional Regression

Nicola Mingotti, Rosa E. Lillo, Juan Romo.

May 9, 2013

*Nicola Mingotti is Phd student in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: nicola.mingotti@uc3m.es). Rosa Lillo is Professor in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: rosaelvira.lillo@uc3m.es). Juan Romo is Professor in the Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid, Spain (E-mail: juan.romo@uc3m.es). This research was supported in part by Spanish Ministry of Education and Science grants MEC 2009/00035/001, ECO2011-25706 and SEJ2007. Our investigations about whiplash injury in car accidents are made in collaboration with *AXA*, *Madrid*. In the specific case of this article, the company provided an important support in mining out of the market car characteristics that would have been difficult to obtain by other means. We want also to acknowledge *Centro Zaragoza* for support in understanding more about car accidents and finally, we acknowledge the *Arbeitsgruppe für Unfallmechanik* for providing their car accidents database and for assistance in our first steps of deployment.

Abstract

Functional Regression has been an active subject of research in the last two decades but still lacks a secure variable selection methodology. Lasso is a well known effective technique for parameters shrinkage and variable selection in regression problems. In this work we generalize the Lasso technique to select variables in the functional regression framework and show it performs well. In particular, we focus on the case of functional regression with scalar regressors and functional response. Reduce the associated functional optimization problem to a convex optimization on scalars. Find its solutions and stress their interpretability. We apply the technique to simulated data sets as well as to a new real data set: car velocity functions in low speed car accidents, a frequent cause of whiplash injuries. By "Functional Lasso" we discover which car characteristics influence more car speed and which can be considered not relevant.

Key Words: Norm one penalization; Variable selection; Algebraic reduction; Convex optimization; Computer algebra.

1. INTRODUCTION

Functional data analysis has received a lot of attention in the last two decades especially after it was popularized by Ramsay and Silverman (2005, 2002). A recent and brief overview can be found in Febrero (2008). A more formal, non parametric approach in Ferraty and Vieu (2006), all references contain practical problems that prove the large applicability of this new technique. We will follow the parametric approach and will describe a variable selection procedure that in our experiments performed well and provided easy to interpret solutions.

A random variable can be considered functional from different points of view. For example, when its realizations can be well fitted by smooth functions. Or, when its inertia can be explained by a function smoothness. Or again, when data can be thought as sampled from underlying unknown smooth functions. Functional regression is a new kind of regression where either the response variable or the regressors are functional data variables. In this work we will focus on the least studied case of a functional linear regression with functional response $Y_i(t)$ and scalar regressors $X_{i,j}$. A case study example about this specific problem can be found in Faraway (1997). A similar problem, where $X_{i,j} \in \{0, 1\}$, is known as FANOVA and has got more attention (Cuevas et al., 2004; Faraway, 2004; Ramsay and Silverman, 2005). The theoretical problems of asymptotics and consistency of estimators have been studied, for the simple linear model, in Cuevas and Febrero (2002).

Given a linear model as:

$$Y_i(t) = \beta_0(t) + \beta_1(t)X_{i,1} + \cdots + \beta_J(t)X_{i,J} + \epsilon_i(t) \qquad i = 1 \ldots I, \quad (1)$$

$\epsilon_i(t)$ represents an error term and has to be thought as a random variable whose distribution is not known in detail[1]. We would like to establish which regressors $\mathbf{X}_\mathrm{j}$ can be considered useful and which can be safely dropped. It is often desirable to drop as many regressors as possible for two reasons. First, a small model is easier to interpret than a large one. Second, its solutions have, in general, a smaller variance. In the case of a classical regression, the response variables Y_i, all parameters β_j and all regressors $\mathbf{X_j}$ are real numbers. There, Lasso has been proven to be a valid technique to perform variable selection and parameters shrinkage (Tibshirani, 1996). An overview of its origins and evolution to more efficient implementations can be found in Hesterberg et al. (2008). Our main contribution in this paper is to adapt the Lasso to our functional case starting from its original definition.

The idea in Lasso is to penalize the absolute magnitude of beta parameters and shrink them until the cross validation error reaches a minimum. Then, discard all estimated parameters $\bar{\beta}_j$ that become too close to zero[2]. In the following we will use a fake regressor $\mathbf{X}_0$ as a vector of ones to improve models readability. With this convention we can write the classical Lasso estimators beginning with:

$$\bar{\beta}_0^{(\lambda)} \dots \bar{\beta}_J^{(\lambda)} = \underset{\beta_0 \dots \beta_J}{Argmin} \sum_{i=1}^{I} \left(Y_i - \sum_{j=0}^{J} \beta_j X_{i,j} \right)^2 + \lambda \cdot \sum_{j=1}^{J} |\beta_j|. \tag{2}$$

Changing λ we find the value $\bar{\lambda}$ that minimizes the cross validation error and call $\bar{\beta}_j$ the Lasso estimators for β_j ($\bar{\beta}_j \stackrel{d}{=} \bar{\beta}_j^{(\bar{\lambda})}$ for all j). Assuming the domain of $Y_i(t)$ functions is $[a, b]$, we can rewrite the Lasso optimization for a functional regression as:

$$\{\bar{\beta}_j^{(\lambda)}(t)\}_j = \underset{\beta_0(t) \dots \beta_J(t)}{Argmin} \sum_{i=1}^{I} \int_a^b \left(Y_i(t) - \sum_{j=0}^{J} \beta_j(t) X_{i,j} \right)^2 dt + \lambda \cdot \sum_{j=1}^{J} ||\beta_j(t)||_1 \tag{3}$$

where $\{\beta_0(t), \dots, \beta_J(t)\}$ live in some function space to be chosen. The 1-norm is defined, as usual, as $||\beta_j(t)||_1 \stackrel{d}{=} \int_a^b |\beta_j(t)| dt$. The last summand in (3) involves the integration of absolute values, therefore obtaining a direct analytic solution of the optimization problem is not trivial. The idea in this paper is to replace the penalization term $\sum_{j=1}^{J} ||\beta_j(t)||_1$ with $\sum_{j,k} |b_{j,k}|$ where $b_{j,k}$ are the coordinates of $\beta_j(t)$ respect to some functional BSpline basis functions $\mathcal{B}$. Show that such a replacement is actually a good approximation to the original constraint and display evidence of its effectiveness on simulated and real data sets. The problem is reduced to an optimization on scalar values by algebraic transformations.

The Lasso, or L_1 regularization, has been already approached in functional regression during the last years. In Matsui and Konishi (2011) it is applied for variable selection to a functional regression with functional predictors and scalar response using Gaussian basis functions. An approach, denoted as *group SCAD*,

permits grouping scalar parameters describing the same functional parameter. In Hong and Lian (2011), the Lasso method is applied to a functional regression with functional predictors and functional response. In this case β_j are scalars and functions are sampled on arbitrary grids to reduce the problem to a numerical solvable one. In Zhao et al. (2012) the response is scalar, there is only one functional variable to estimate, the basis is Wavelet and Lasso is used to set to zero as many coefficients as possible in the Wavelet basis. In this paper we consider the case of multiple functional linear regression with functional response and scalar predictors. We have to estimate several $\beta_j(t)$ parameters and recognize which ones can be considered zero. We use the traditional BSpline basis functions to describe the functional objects. The functional optimization problem is reduced to a numerical optimization problem with basic algebraic manipulations, we do not need to sample over arbitrary grids of values. Finally, the focus is on functional variable selection, not on single scalar parameters going to zero. We have found no need to cluster explicitly scalar parameters pertaining to the same functional object. BSplines provide this implicitely. Indeed, BSpline basis functions are not fully orthogonal, each of their internal parameters is related to its closest neighbours (de Boor, 2001; Iglesias et al., 2007). Then, if a parameter goes to zero, its neighbours will be affected and tend to get small values. Since all internal parameters are chained by their respective neighbours, then a property of the majority of coefficients tends to become a property of all the coefficients. In conclusion, if many coefficients are zero for a functional object described in a BSpline basis, then the whole functional object tends to become zero.

The case of scalar regressors and functional response is, in some sense, more troublesome than others because it requires to control a function by a set of scalar valued regressors that are lower dimensional and less information rich. The error terms ϵ_i are in this context functions $\epsilon_i(t)$, we can not apply familiar distributional properties. Finally, the results of our models will be functions. To understand if they are working, we have to compare them with other output functions; that is, we have to compare plots. This is impractical, time consuming and error prone but is the only serious option.

We applied our *Functional Lasso* method to study a new real data set. We have a database of low speed car accidents that mimicks the tipical scenario of whiplash injury. We want to explain/predict the speed function of an impacted car just after the impact knowing some of the two cars characteristics as car weight, height, length, speed difference, etc. Predicting the speed function is the first step to better understanding the whiplash injury. Whiplash is a very common injury and has severe repercussions on society. Besides the physical pain and discomfort of injured people, its economical costs are quite remarkable. It has been estimated that in the U.S. whiplash costs annually $29 billions. Its incidence is approximately 4 per 1000 persons (Eck and Hodges, 2001). The relation between car dynamics and whiplash injury risk has been studied in Kraft et al. (2011). There, it is shown that speed difference and average acceleration are correlated with injury severity and symptoms duration. With this article

we intend to start a new branch of research into this topic. For the first time, instead of using mean velocities and accelerations we use complete functional representations of the variables of interest, in particular $v(t)$.

The paper contains four more sections. In Section 2 we are going to detail what is the algebraic form of the optimization problem and justify the passage from $||\beta_j||_1$ to $\sum_{j,k} |b_{j,k}|$ as direct consequence of BSpline properties. In Section 3 we show the method performance on simulated data sets. In Section 4 we apply the method to the car accidents database.

2. METHODOLOGY ILLUSTRATION

Problem (3) is a functional problem that could be difficult or very tedious to solve analytically. To be able to compute numerically $\bar{\beta}_j^{(\lambda)}(t)$, we choose a basis function $\mathcal{B} = \{\phi_0(t), \ldots, \phi_K(t)\}$ and express all functions in (3) as a linear combination of the basis. For example, the response variables become $Y_i(t) = \sum_{k=0}^{K} a_{i,k}\phi_k(t)$ and beta parameters become $\beta_j(t) = \sum_{k=0}^{K} b_{j,k}\phi_k(t)$. It has to be stressed that coefficients $a_{i,k}$ are known real numbers because $Y_i(t)$ are known functions. On the contrary, $b_{j,k}$ are unknown reals since $\beta_j(t)$ are unknown functional parameters to be estimated. We will estimate the values of $b_{j,k}$ solving the following optimization problem and denote the estimates as $\bar{b}_{j,k}^{(\lambda)}$.

$$\{\bar{b}_{j,k}^{(\lambda)}\}_{j,k} = \underset{b_{j,k}}{Argmin} \sum_{i=1}^{I} \int_a^b \left(\sum_{k=0}^{K} a_{i,k}\phi_k(t) - \sum_{j=0}^{J} (\sum_{k=0}^{K} b_{j,k}\phi_k(t))\ X_{i,j} \right)^2 dt + \\ + \lambda \cdot \sum_{j=1}^{J} \int_a^b \left| \sum_{k=0}^{K} b_{j,k}\phi_k(t) \right| \ dt \ . \tag{4}$$

The first part of the optimization function, the sum of integrals of a square, reduces algebraically to a quadratic form on variables $b_{j,k}$ but the remaining part can not be easily simplified without further informations. To overcome this difficulty we resort to a BSpline property, citing DeBoor *"B-Spline coefficients model the function they represent."*, see de Boor (2001), Example IX.2. The property is illustrated by an example in Figure 1. If we suppose we are using a cubic spline with knots $\{t_0, t_1, \ldots t_n\}$ on the domain $[a, b]$ where $t_0 = t_1 = t_2 = t_3 = 0,\ t_n = t_{n-1} = t_{n-2} = t_{n-3} = 1$ and for all other t_i we set $\Delta := t_{i+1} - t_i$. Then, for each j:

$$\int_a^b \left| \sum_{k=0}^{K} b_{j,k}\phi_k(t) \right| dt \;= ||\beta_j(t)||_1 \;=\; \int_a^b |\beta_j(t)|\; dt$$
$$\approx \sum_{i=3}^{n-4} |\beta_j(t_i)| \cdot \Delta \quad (Riemann\ Integral)$$
$$\approx \Delta \cdot \sum_{k=1}^{K-2} |b_{j,k}| \;\leq\; \Delta \cdot \sum_{k} |b_{j,k}| \quad (BSplines\ Property) \tag{5}$$

The Δ value can be removed because it would only rescale λ (see Eq. 4). We finally get the objective function:

$$\{\bar{b}_{j,k}^{(\lambda)}\} = \underset{b_{j,k}}{Argmin} \left(Quadratic(b_{j,k}) + \lambda \cdot \sum_{j,k} |b_{j,k}| \right). \tag{6}$$

Fixing λ ($\lambda \geq 0$) we can easily compute (6) because it is now a numerical convex optimization problem, (a proof con be found in the Appendix B, Proposition 1) for which there are specialized solvers as CVX (Grant and Boyd, 2011, 2008). The problem being convex ensures solutions $\bar{b}_{j,k}^{(\lambda)}$ to be unique.

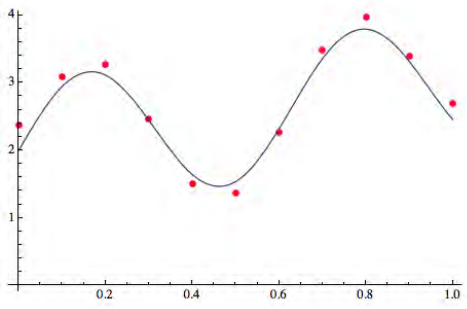

Figure 1: fig:the-bspline-property

3. SIMULATED DATA

In this section we will create two artificial data sets with known linear models. Then, we will apply the Lasso technique to rediscover the underlying models. The first example is to show how Lasso discriminates correctly the original regressors from fake ones. The second shows that Lasso, in absence of fake regressors to drop, reduces to least squares.

3.1 Functional Lasso annihilates spurious regressors

The data set is made of 30 functional observations built as follows.

- Generate 6 regressors as 6 vectors of 30 random values: $X_{i,j} \sim N(0, \sqrt{5}),\ i = 1 \dots 30,\ j = 1 \dots 6$.[3]
- Define three functions $\beta_j(t),\ t \in [0,1], j = 0, 1, 2$ as:

$$\begin{cases} \beta_0 = 30t(1-t)^{3/2} \\ \beta_1 = 10(t-0.6)^2 + 1 \\ \beta_2 = Sin(4\pi t) - 2\ . \end{cases} \tag{7}$$

- Define 30 error functions $\epsilon_i(t)$ in $t \in [0,1]$ by generating 101 random points P_k and then joining them continuously with a linear interpolation. $P_k := (x_k, y_k)$, $x_k := \frac{1}{100}\ k$, $y_k \sim Normal(0, 0.8)$ for $k = 0, 1, \dots, 100$. The value $\sigma = 0.8$ is arbitrary but appears to be reasonably sized (see Figure 3).
- Generate 30 functional response variables as:

$$y_i(t) = \beta_0(t) + \beta_1(t)X_{i,1} + \beta_2(t)X_{i,2} + \epsilon_i(t) \qquad i = 1 \dots 30 \tag{8}$$

- Get the discrete representation of the response variables as: $Y_{i,j} = y_i(t_j)$, $t_j = \frac{1}{100}j,\ j = 0, 1, \dots, 100$. From this point on consider cleared the variables $y_i(t)$, we need them to denote other objects. For a representation of $y_i(t)$ and Y_i see Figure 2.

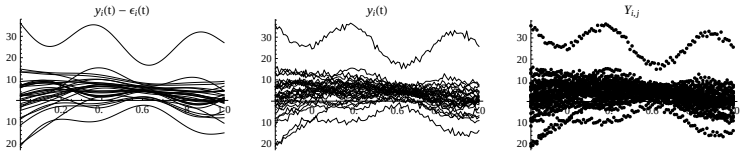

Figure 2: fig:generate-fake-Y

At this point we have a data set $(Y_{i,j}, X_{i,1}, X_{i,2}, X_{i,3}, X_{i,4}, X_{i,5}, X_{i,6})$ where $i = 1 \dots 30$, $j = 0 \dots 100$. We forget we know how this data has been generated. To improve readability we will use a vector notation to denote the discrete response variables. $\mathbf{Y}_i$ will be a vector of 101 elements whose k-th element is $Y_{i,k}$. We want to explain the response variable $\mathbf{Y}_i$ by mean of a functional linear regression model with functional response and scalar covariates $\{X_{i,1}, X_{i,2}, X_{i,3}, X_{i,4}, X_{i,5}, X_{i,6}\}$. A well performing method will recognize that useful regressors are only $\{X_{i,1}, X_{i,2}\}$ and will find the estimated parameters $\tilde{\beta}_0(t), \tilde{\beta}_1(t), \tilde{\beta}_2(t)$ [4] to be close to the original parameters $\beta_0(t), \beta_1(t), \beta_2(t)$.

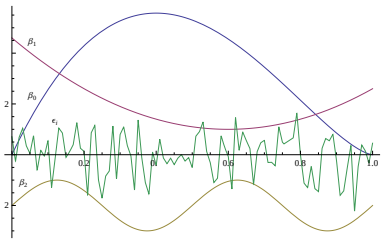

Figure 3: fig:initial-params

As required to apply functional regression we transform the response variables $\mathbf{Y}_i$ into functions. One way to do this is to fit each $\mathbf{Y}_i$ with a function in some predefined functions space. The choice of the functions space is in part arbitrary, Ramsay and Silverman (2005) present some classical basis functions and some rules of thumb to choose between them. In this case we choose order three BSpline basis functions with ten equally spaced internal knots more three equal knots at points 0 and 1, beginning and end of functions domain. The equal values at the ends are required to reduce smoothness at the domain borders.

$$\mathcal{K} = \{0, 0, 0, 0, 0.1, 0.2, \ldots, 0.8, 0.9, 1, 1, 1, 1\} \tag{9}$$

The knots sequence has been chosen by trading off simplicity and effectiveness. Other ways to place the knots are surely possible. A cross validation could be used to determine, in some sense, the optimal number of knots, but the amount of smoothness required for each case study remains largely dependent on the eye of the modeler (Faraway, 1997). Using the knots sequence $\mathcal{K}$ we obtain a 13 element cubic BSpline basis functions $\mathcal{B} = \{\phi_0(t), \phi_1(t), \ldots \phi_{12}(t)\}$, we compute them with *Mathematica 8.0 BSplineBasis* built-in command. An introduction to symbolic BSplines manipulation with *Mathematica* can be found in Iglesias et al. (2007). The analysis proceeds as follows:

- We standardize all variables. Standardized regressors will be denoted with $\mathbf{XS}_j$ and computed naturally as:

  $$\mathbf{XS}_j := \mathit{Standardize}((X_{1,j}, X_{2,j}, \ldots X_{30,j})) \quad \mathit{for}\ j \in \{1, \ldots, 6\}, \tag{10}$$

  response variables $\mathbf{Y}_i$ are standardized all together as:

  $$\mathit{Standardize}((Y_{1,0}, \ldots, Y_{1,100}, Y_{2,0}, \ldots, Y_{2,100}, \ldots, Y_{30,0}, \ldots, Y_{30,100})), \tag{11}$$

  their standardized versions are denoted $\mathbf{YS}_i$.

- Fit each $\mathbf{YS}_i$ to a function $y_i(t)$ in the functions space determined by $\mathcal{B}$ minimizing the squared error.

- Set $\mathbf{XS}_0$ to be a length 30, vector of ones and and define a linear model $\mathcal{M}_i(t)$ to explain each of $y_i(t)$ as $\mathcal{M}_i(t) := \sum_{j=0}^{6} \beta_j(t) \cdot XS_{i,j}$. [5]

- A solution in this context is an estimation of the $\beta_j(t)$ parameters giving a best fit to the data. We find here two kinds of solutions and compare them, the common Least Squares solution and the new Functional Lasso solution, each of them will be denoted respectively as LS and FL. Parameters that are LS solution will be denoted as $\hat{\beta}_j(t)$, FL solution will be denoted as $\bar{\beta}_j(t)$.

$$\{\hat{\beta}_j\}_{j=0\dots6} := \underset{\{\beta_j\}_{j=0\dots6}}{Argmin} \sum_{i=1}^{30} \int_0^1 (y_i(t) - \mathcal{M}_i(t))^2 \tag{12}$$

$$\{\bar{\beta}_j^{(\lambda)}\}_{j=0\dots6} := \underset{\{\beta_j\}_{j=0\dots6}}{Argmin} \sum_{i=1}^{30} \int_0^1 (y_i(t) - \mathcal{M}_i(t))^2 + \lambda \cdot \sum_{j=1\dots6} ||\beta_j(t)||_1 \tag{13}$$

- Observe explicitly that $\hat{\beta}_j = \bar{\beta}_j^{(0)}$, then we are computing them in this way. We are going to reduce the sum of integrals to a quadratic forms in $b_{j,k}$ by means of *Mathematica* computer algebra capabilities. Reduce $\sum ||\beta_j(t)||_1$ to $\sum |b_{j,k}|$ as illustrated in the previous section and solve the resulting unconstrained convex optimization problem $\underset{b_{j,k}}{Argmin}\ Quadratic(b_{j,k}) + \lambda \cdot \sum |b_{j,k}|$ by *Matlab CVX* package. The part that takes more time is the algebraic reduction of integrals, more or less half an hour with a mid-range laptop, the optimization part is faster and takes around a minute.

- We compute $\bar{\beta}_j^{(\lambda)}$ for many values of λ and look for the value $\bar{\lambda}$ that minimizes the five-out cross validation error. Once found, we define the FL parameters as $\bar{\beta}_j(t) := \bar{\beta}_j^{(\bar{\lambda})}$ for $j = 0 \dots 6$.

The estimated parameters computed by least squares ($\hat{\beta}_j(t)$) and by the Functional Lasso ($\bar{\beta}_j(t)$) can be seen in Figure 4 and Figure 5. In Figure 4 there is a direct comparison between $\hat{\beta}_j(t)$ (in dashed red stroke) and $\bar{\beta}_j(t)$ (in full black) for each j. It can be seen that Lasso shrinks all spurious parameters $\{\beta_3(t), \dots, \beta_6(t)\}$ to zero while ordinary least squares keep them fluctuating around the x-axis without annihilating them. It is exactly the same thing that happens in ordinary multiple regression. The difference here is that, instead of scalars, whole functional parameters are set to zero. It is much easier to decide which is a useless regressor using Lasso solution. The shape of estimated $\{\beta_0, \beta_1, \beta_2\}$ are similar to their original values for both methods, only at domain borders there is a little discrepancy. The cross validation is minimal for the Lasso solutions, we shrunk the parameters but we actually improved the performance of the model. We must stress that parameter selection in this case has been very easy since some of them have been completely shut down to zero. In general, it will not always be so clear, therefore we set a formal rule to decide if a parameter has to be dropped.

Fact 1. Rule of magnitude. We consider a regressor variable X_j spurious, or not effective, if its associated functional parameter is too small in magnitude:

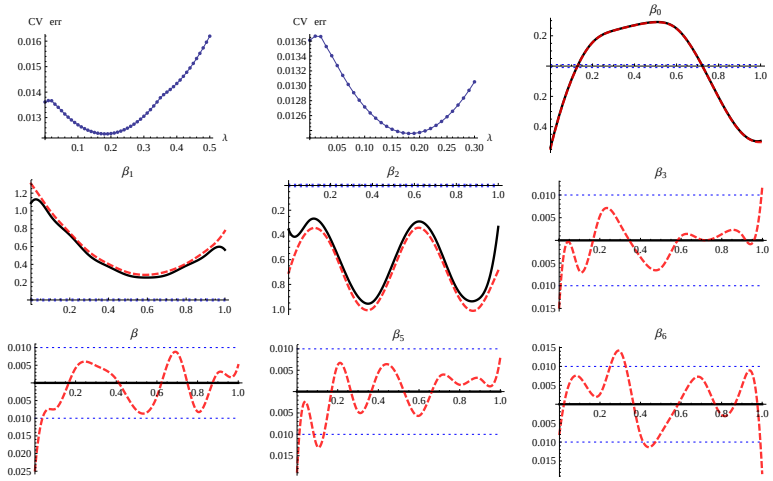

Figure 4: fig:plate-1

$\underset{t\in[0,1]}{max}\ |\beta_j(t)| \leq 0.01$. This rule is a conservative extension to the functional context of the one implicitly used in Tibshirani (1996). It makes sense only when variables are standardized or transformed to lay near zero. Parameter 0.01 is arbitrary.

In Figure 5 the parameter shrinkage process is shown. We change the penalization term λ in the interval $[0, 0.02]$ and observe how parameters do change. It is manifest that spurious parameters change a lot. On the contrary, effective parameters remain almost unvaried. In this case we know in advance which parameter should be dropped but, in general, this could be a useful explorative technique to decide if a parameter has to be retained or dropped. If, increasing λ, some parameters change far more than others then these parameters are likely spurious parameters.

Fact 2. Rule of inertia. Increasing the penalization term λ, in a functional Lasso, shrinks and changes the shape of parameters $\beta_j(t)$. The most inertial parameters, the ones who change less while increasing λ, tend to be the most influential ones.

The result synthesised in the last *Rule of inertia* has been observed during experimentations with different linear models and error function realizations. In this paper it can be seen applied again on the real data set, see Figure 14.

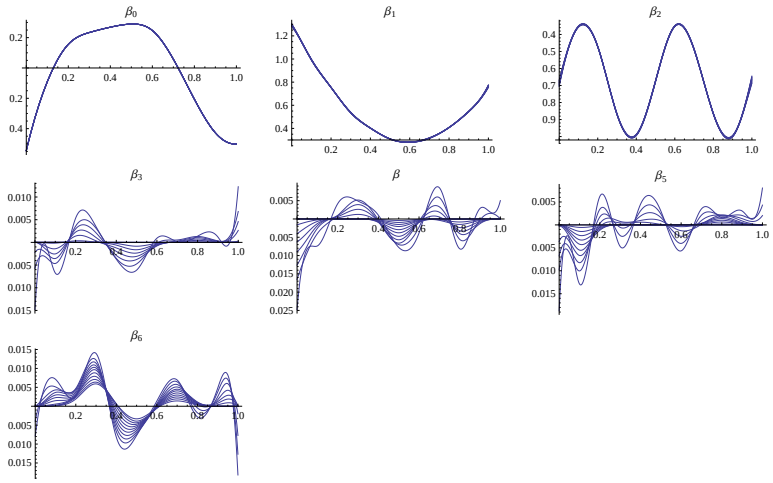

Figure 5: fig:plate-2

3.2 Functional Lasso does not annihilate useful regressors

What happens if there are not spurious regressors? Does Lasso try to drop the useful ones? The answer is no. In case all parameters are useful, Lasso selects all of them and reduces to the LS solution. An example can be realized using the same method of the previous simulation with a few changes.

- Define four functions $\beta_j(t),\ t \in [0,1],\ j = 0,1,2,3$, as:

$$\begin{cases} \beta_0 = 30t(1-t)^{3/2} \\ \beta_1 = 10(t-0.6)^2 + 1 \\ \beta_2 = Sin(4\pi t) - 2 \\ \beta_3 = Cos(4\pi t + 0.5) + 1\ . \end{cases} \tag{14}$$

- Generate 30 functional response variables depending on three regressors

$$y_i(t) = \beta_0(t) + \beta_1(t)X_{i,1} + \beta_2(t)X_{i,2} + \beta_3(t)X_{i,3} + \epsilon_i(t) \qquad i = 1 \dots 30 \tag{15}$$

- Reduce functions $y_i(t)$ to numerical observations $\mathbf{Y}_i$ by sampling, then standardize $\mathbf{Y}_i$ and $X_{i,j}$ regressors.

- Apply the Functional Lasso technique to the data set $(\mathbf{Y}_i, X_{i,1}, X_{i,2}, X_{i,3})$ for $i = 1 \dots 30$. Observe this time we have exactly the same regressors we used in the model. If the technique performs well it has to recognize that all regressors are useful and rebuild the parameters $\beta_0(t) \dots \beta_3(t)$ as best as possible.

The cross validation error is represented in Figure 7. It is monotonically increasing as λ increases and the minimum is at $\lambda = 0$. Then, the solution reduces to least squares. $\beta_j(t)$ shapes are correctly estimated, as can be seen in Figure 6. Their differences in scale are a consequence of standardization. This result has occurred repeatedly in our experiments, so we conjecture the result holds in general and spell it as a rule.

Fact 3. Reduction to LS. In case there will be no regressors to drop Lasso method will choose, as best λ, the value $\bar{\lambda} = 0$ and FL solution will reduce to the LS solution.

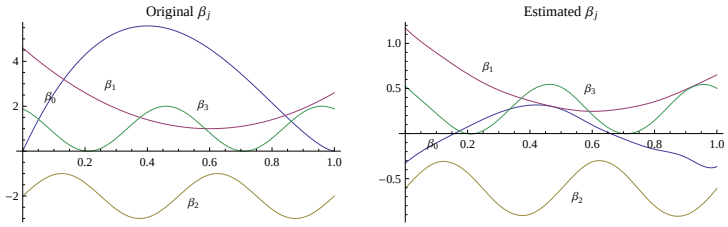

Figure 6: fig:3-reg-betas

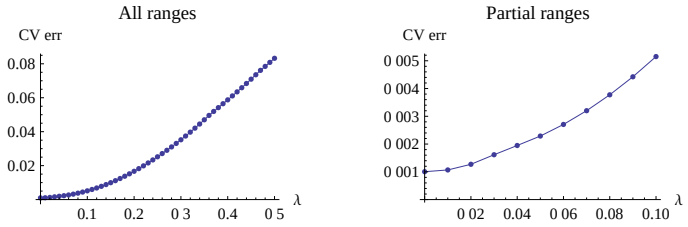

Figure 7: fig:3-reg-CV

4. CASE STUDY: LOW SPEED CAR ACCIDENTS AND WHIPLASH INJURY

In this section we are going to use the Functional Lasso to predict the velocity of an impacted car in a low speed accident. This study is part of a larger project for the understanding and control of whiplash injury risk.

Whiplash injury is very common and its incidence is about 4 per 1000 persons. It happens when sudden acceleration-deceleration forces are applied to the neck and the upper trunk. The term "whiplash" was introduced in 1928. Before, the injury was referred to as "railway spine" since the most frequent cause of it were train accidents. Nowadays, the most frequent cause are car accidents.

Victims are usually sitting in a car standing still when another car hits it in the back. Whiplash injuries are usually not life threatening but they are common, expensive and can give long term consequences. It has been estimated that the U.S. annual economic cost related to whiplash is $3.9 billion, including medical care, sick leave and lost work productivity. Taking into account also litigation costs the number rises to $29 billion (Eck and Hodges, 2001).

Whiplash risk is correlated with the impacted car speed variation and its average acceleration (Kraft et al., 2011). In the following we will try to predict the impacted car speed function $v(t)$. Half of the data base we are using is publicly available, in raw form, at *AGU (Arbeitsgruppe für Unfallmechanik)*. The other half comes from proprietary *AXA* documentation. *AGU* data contains high frequency speed and acceleration measurements for a set of more than a hundred car accidents. For each car in each crash we extracted some car charatheristics from AXA documentation resources as car weight, length, etc.

From all the car accidents we selected a set of 25 that are particularly homogeneous. In each selected accident there are two cars, A and B. Car B is initially motionless. Car A is initially traveling at some known constant low[6] speed until it hits car B in its back. Cars A and B are perfectly aligned: from the top view their symmetry axes lay on the same line. Car B does not have the rear hook. For each car we have available the following variables: initial speed (vi), weight (wei) , length (len), width (wid), height (hei), and we know the speed, as a function of time, of car B after it has been hit ($v^{(B)}(t)$). Our aim is to model and predict $v^{(B)}(t)$ for the first 0.2 seconds after the impact, from the impact characteristics.

The problem can be seen as functional linear regression. The response variable $v^{(B)}(t)$ is functional and $\{$ vi, wei, len, wid, hei $\}$ are scalar regressors. Instead of using directly these regressors, mechanics considerations suggest we use their standardized differences $\{$ $AviS$, $\Delta weiS$, $\Delta lenS$, $\Delta widS$, $\Delta heiS$ $\}$. For example, $\Delta weiS$ is the standardized vector of differences in weight between car B and A, $\Delta lenS$ is the standardized vector of length differences and so on for all other variables. The only exception is $AviS$, since Bvi is always zero, we only standardized car A speeds. Correlations between regressors are shown in Table 1.

Table 1: Regressor correlations for the car speeds problem.

	$AviS$	$\Delta weiS$	$\Delta heiS$	$\Delta widS$	$\Delta lenS$
AviS	1.00	0.61	0.16	0.61	0.44
$\Delta weiS$	0.61	1.00	0.16	0.87	0.81
$\Delta heiS$	0.16	0.16	1.00	0.05	-0.23
$\Delta widS$	0.61	0.87	0.05	1.00	0.84
$\Delta lenS$	0.44	0.81	-0.23	0.84	1.00

Each response variables $v_i^{(B)}(t)$ is originally represented as a set of (x, y) coordi-

nates of varying length. We rescale the x coordinate to $[0,1]$ interval and standardize respect to y. This means we standardize a curve speed value respect to all 25 curve speed values. Then, we approximate each curve points with a Spline function minimizing the least square error. In Figure 8, plot (a) are represented the original car accelerations. In plot (b) car velocites. Finally, in plot (c) the standardized BSplines smoothed velocities we will use in our functional regression. The BSpline basis is the same used in the simulated data example, order 3 with equally spaced knots sequence: $\mathcal{K} = \{0,0,0,0,0.1,0.2,\ldots,0.8,0.9,1,1,1,1\}$. The basis is chosen for its simplicity.

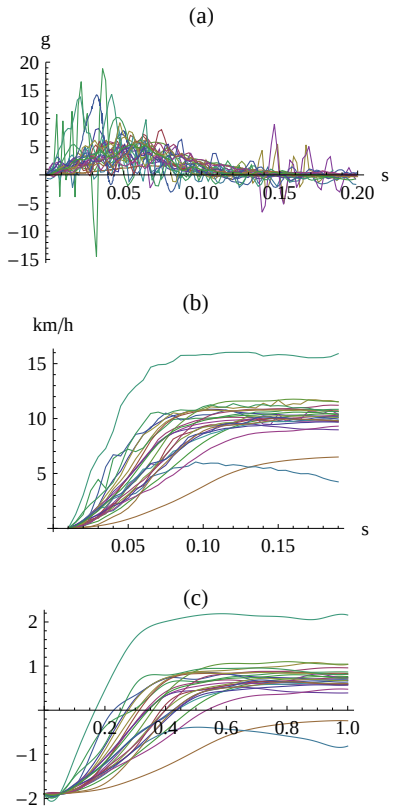

Figure 8: fig:curves-acc-vel

Model 0. By the same procedure illustrated in the previous section we find the best parameter estimation $\hat{\beta}_0(t),\ldots,\hat{\beta}_5(t)$ for the linear model

$$\begin{aligned} v_i(t) = \beta_0(t) + \beta_1(t)\ AviS + \beta_2(t)\ \Delta weiS + \beta_3(t)\ \Delta lenS + \\ \beta_4(t)\ \Delta widS + \beta_5(t)\ \Delta heiS + \epsilon_i(t). \end{aligned} \tag{16}$$

Compare two kinds of solutions, the ordinary one given by least squares (LS) with the new one provided by functional Lasso (FL). All solutions are obtained working with an 11-out cross validation sample, almost half of the set. The first 11 curves of the set are left out and considered test set, see Figure 9 for an illustration. The first FL solution we obtain is not practically useful but interesting. Looking at Figure 10 we see that there is a minimum in the cross validation error ($\lambda \approx 0.82$) but globally that minimum gives a very small gain respect to larger values of λ. So, the prediction error is not notably small compared to the one of a trivial model containing only $\beta_0(t)$. In Figure 11 we can see parameters estimated by LS in dashed red, FL in solid green. According to the *Rule of magnitude*, LS accepts all parameters.Δ *heiS* and *AviS* are smaller than the other ones but not always less than 0.01. On the contrary, FL rejects (sets to zero) all parameters exceptΔ *widS*. FL solution is already better respect to LS because it is more compact, only one regressor has been selected and the cross validation error is smaller. This solution is not very informative because the cross validation error is very near to the one at $\lambda \to \infty$ and last, but not least, the only variable selected isΔ *widS*, this clashes with our physical intuition.

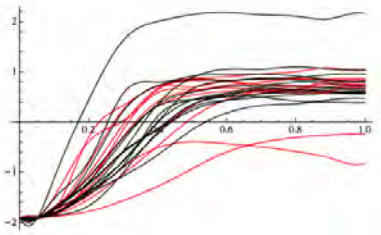

Figure 9: fig:cross-in-out-real

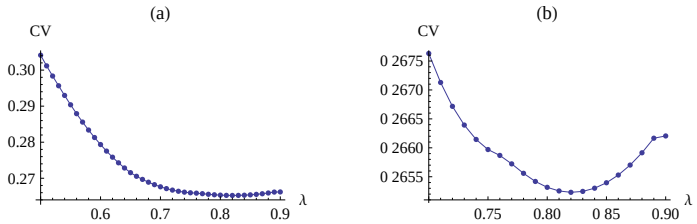

Figure 10: fig:CV-con-width

Model 1. Looking at Table 1 it is easy to see what happened,Δ *widS* is highly correlated withΔ *weiS* andΔ *lenS*. The weight, an expected dominant variable in every dynamics problem has been shaded by another, linearly correlated but much humbler. We prefer weight to be in our model respect to width, so we annihilateΔ *widS* setting a constraint in the optimization phase: $\sum_{k=0}^{12} |b_{5,k}| \leq$

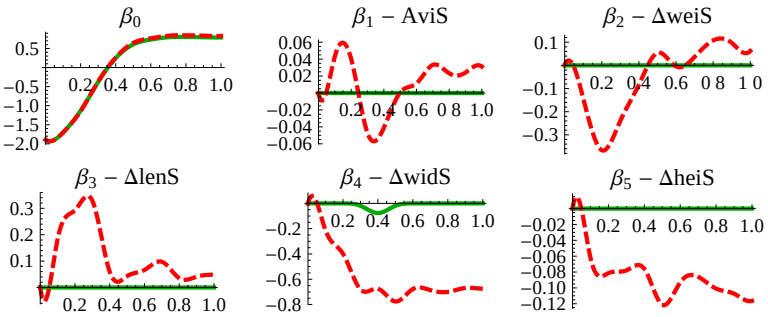

Figure 11: fig:betaHAT-con-width

10^{-7}. Then, we estimate again the LS and FL parameters. From Figure 12 we observe preliminarily how the cross validation error minimum is now one order of magnitude deeper. The CV minimum at $\lambda \approx 0.50$ is better than LS ($\lambda = 0$) and also better than the trivial model ($\lambda \to \infty$). Next, observing Figure 13 we see that FL has dropped two variables, { $\Delta heiS$, $\Delta lenS$ } and shrunk the other two, {$AviS$, $\Delta weiS$ }. We can conclude that FS solution is better than LS because it is simpler (it has fewer regressors) and has stronger predictive power (smaller cross validation error).

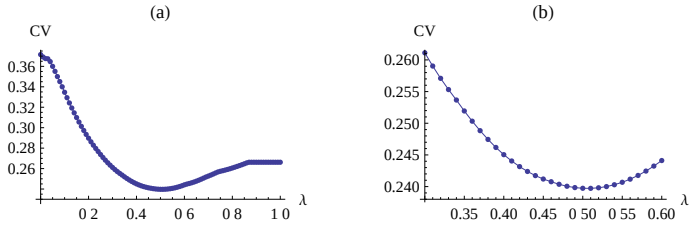

Figure 12: fig:CV-no-width

Model 2. In our curve set there are three elements that could be considered unusual or outliers. Looking at Figure 9 we can clearly see them. One is the curve reaching highest values, the other two are always laying below the x-axis. We removed these observations and performed again LS and FL with 5-out and 10-out (half-out) cross validation. LS still selects all variables while FL indicates now there is no variable worth keeping. FL reaches the minimum cross validation error when all parameter values become extremely small.

Model 3. Retaining outliers as observations but using them only in the cross validation part (half-out cross validation). LS still selects all variables. FL selects $AviS$, $\Delta weiS$ and drops the others.

We can conclude that only two variables can not be discarded if we take into account all the data we have accumulated, the speed and weight of the two

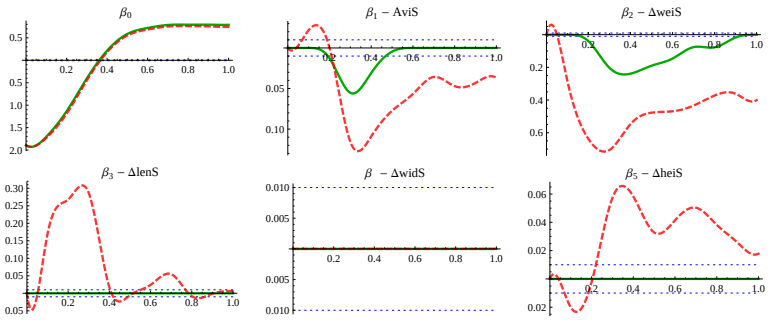

Figure 13: fig:betaHat-no-width

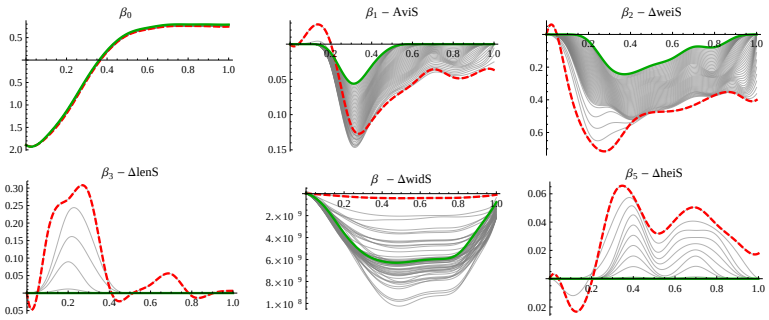

Figure 14: fig:beta-change-no-width

cars. This is in agreement with the classical mechanics notion of momentum and the law of conservation of mechanical energy. The estimated parameters in Figure 13 display also that car A velocity has an influence in time only between 0.2 and 0.4 (standardized time units) whileΔ wei is influential for a much larger time period. The sign of $AviS$ should not cause confusion, if that variable would have been standardized as all others we would have had $std(Bvi - Avi) = std(-Avi) = -AviS$ so the function would have appeared reversed and ultimately, more appealing to intuition. The large outlying value is for a crash where car A was 650 Kg heavier than car B [7]. The two small outliers have different characterizations. The one in which $v_B(t)$ decays in the right part of the plot is for two cars with very similar weight ($\Delta wei = 21Kg$), the other is a case in which car B is far heavier than car A (588 Kg more). This confirms (or implies) the parameters analysis in Figure 13,Δ $weiS$ has a durable impact, in time, on $v_B(t)$.

5. CONCLUSIONS

In this paper we presented a new method for variables selection in functional regression with functional response and scalar regressors. The method is an extension to the well known Lasso technique to the functional case. We applied the FL (Functional Lasso) to artificial datasets as well as to a new real data set. The results are very promising. On the artificial data sets the FL procedure discriminated active regressors from fake ones. Moreover, the estimated functional parameters turn out to be easily interpretable, they do not display confusing humps common in fuctional regression solutions. Three phenomenological rules are set to guide the general process of variable selection: *Rule of magnitude*, *Rule of inertia* and *Rule of reduction to Least Squares*.

As a real data set benchmark we studied low speed car accidents, a frequent whiplash injury cause. We related the speed function of the struck car to the initial difference in speed between the two cars, their weight, their height, their width and length. Studying a set of 25 accidents we can conclude that the only two variables can be considered significant: the weight difference and the pre-impact speed difference. The weight difference has a more durable effect in respect to the speed difference.

The choice of BSpline basis gives many benefits. In functional data analysis we suppose our data consists of noisy samples from some underlying, inaccessible functions. With BSplines we can roughly control these functions variability, in their domain, while defining the knots sequence. They allow the description of non periodic functions. BSpline coefficients approximate the fitted function values and this permitted us to approximate each $||\beta_j(t)||_1$ with $\sum_k b_{j,k}$ that is, to solve a Lasso on functional objects by a Lasso on scalars. Lastly, BSpline basis functions are not orthogonal, this at first seems a negative characteristic but it is what gives our estimators interpretability. Indeed, if many coefficients of a functional object go to zero, then they tend to pull to zero all their neighbours, that is all other coefficients of the same object.

APPENDIX A: FOOTNOTES

[1]We are assuming $\epsilon_i(t)$ has, in some sense, mean zero.

[2]Implicitly, Tibshirani (1996) defines "too small" as smaller, in absolute value, than 0.01.

[3]Normal distribution will be always written as $N(\mu,\sigma)$.

[4]Here $\tilde{\beta}_j(t)$ denotes the estimation of a parameter $\beta_j(t)$ by some method left to determine.

[5]$XS_{i,j}$ is the i-th element in vector $\mathbf{XS_j}$.

[6]Low speed here means a speed inferior to 30 Km/h.

[7]Consider an average a car in Spain weights approximately 1250 Kg. Figure computed on 1200 cases whiplash accident closed by AXA in 2011.

APPENDIX B: CONVEXITY PROOF

Lemma 1. *If $\{f_1, \ldots, f_n\}$ are convex functions and $\{w_1, \ldots, w_n\}$ are non negative real numbers then $\sum_{i=1}^{n} w_i f_i$ is a convex function (Boyd and Vandenberghe, 2004).*

Proposition 1. *Problem (6) is a convex optimization problem.*

Proof. For an optimization problem to be a convex optimization problem the objective function and the constraints have to be convex functions (Boyd and Vandenberghe, 2004). In expression (6) there are no constraints, the domain is the whole $\mathbb{R}^{K+J+2}$, so we have only to check that $(Q(\boldsymbol{b}) + \lambda \cdot \sum |b_{j,k}|)$ is a convex function. $Q(\boldsymbol{b})$ is a quadratic form, a quadratic form is a convex function iff it is positive semidefinite that is, iff $Q(\boldsymbol{b}) \geq 0$, for all $\boldsymbol{b}$.

$$Q(\boldsymbol{b}) := \sum_{i=1}^{I} \int_0^1 \left(\sum_{k=0}^{K} (a_{i,k} - \sum_{j=0}^{J} b_{j,k} X_{i,j}) \phi_k(t) \right)^2 dt \quad \geq 0, \quad for\ all\ \boldsymbol{b} \tag{17}$$

Whatever $\boldsymbol{b}$ is chosen as argument, $Q(\boldsymbol{b})$ is a sum of integrals of non-negative functions which implies it is always a non-negative value. The absolute value is a convex function, $|b_{i,j}|$ is convex. Then, using Lemma(1) we get that $\sum |b_{i,j}|$ is convex and, remembering $\lambda \geq 0$, also that $Q(\boldsymbol{b}) + \lambda \sum |b_{i,j}|$ is convex. □

References

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Cuevas, A. and Febrero, M. (2002). Linear functional regression: The case of fixed design and functional response. *Canadian Journal of Statistics*, 30(2):285–300.

Cuevas, A., Febrero, M., and Fraiman, R. (2004). An anova test for functional data. *Computational statistics & data analysis*, 47:111–122.

de Boor, C. (2001). *A practical guide to splines*. Springer-Verlag.

Eck, J. and Hodges, S. (2001). Whiplash: a review of a commonly misunderstood injury. *The American journal of medicine*, 110:651–656.

Faraway, J. (1997). Regression analysis for a functional response. *Technometrics*, 39(3):254–261.

Faraway, J. (2004). An F test for linear models with functional responses. *Statistica Sinica*, 14:1239–1257.

Febrero, M. (2008). A present overview on functional data analysis. *Boletín de Estadística e Investigación Operativa. BEIO*, 24(1):6–12.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer.

Grant, M. and Boyd, S. (2008). *Graph implementations for nonsmooth convex programs*. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited. `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, M. and Boyd, S. (2011). *CVX: Matlab Software for Disciplined Convex Programming, version 1.21*. `http://cvxr.com/cvx/`.

Hesterberg, T., Choi, N., Meier, L., and Fraley, C. (2008). Least angle and l1 penalized regression: A review. *Statistics Surveys*, 2:61–93.

Hong, Z. and Lian, H. (2011). Inference of genetic networks from time course expression data using functional regression with lasso penalty. *Communications in Statistics - Theory and Methods*, 40:1768–1779.

Iglesias, A., Ipanaqué, R., and Urbina, R. (2007). Symbolic manipulation of bspline basis functions with mathematica. *Computational Science–ICCS 2007*, pages 194–202.

Kraft, M., Anders, K., Malm, S., and Ydenius, A. (2011). Influence of Crash Severity on Various Whiplash Injury Symptoms: A Study Based on Real-Life Rear-End Crashes with Recorded Crash Pulses. *Folksam Research and Karolinska Institutet, Sweden*.

Matsui, H. and Konishi, S. (2011). Variable selection for functional regression models via the L1 regularization. *Computational Statistics & Data Analysis*, 55:3304–3310.

Ramsay, J. and Silverman, B. (2002). *Applied functional data analysis: methods and case studies*. Springer.

Ramsay, J. and Silverman, B. (2005). *Functional data analysis*. Springer.

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

Zhao, Y., Ogden, R. T., and Reiss, P. T. (2012). Wavelet-based lasso in functional linear regression. *Journal of Computational and Graphical Statistics*, 21(3):600–617.

CAPTIONS

- **Figure 1.** A function $f(x) = 2 + x + Sin(10 \cdot x)$ in $[0, 1]$ has been plotted in blue, it has been sampled in 101 points $x_i = i \cdot 0.01$ for $i = 0, \ldots, 100$ and finally fitted by an order 3 BSplines with 17 knots $\mathcal{K} = \{0, 0, 0, 0, 0.1, 0.2, \ldots, 0.8, 0.9, 1, 1, 1, 1\}$. The initial and final repeating values are needed for $f(x)$ non periodicity. The BSpline functions basis is composed of 13 elements $\{\beta_0(t), \ldots, \beta_{12}(t)\}$. To that basis $f(x)$ is represented as $\sum_{i=0}^{12} b_i \beta_i(t)$. The red points have coordinates $(x_i, y_i) := (0.1 \cdot (i-1),\ b_i)$ for $i = 1 \ldots 11$, and lay all tight around the function graph. We could include extreme basis coefficients averaging.

- **Figure 2.** Generation of an artificial data set $Y_{i,j}$ by known functional parameters $\{\beta_0(t), \beta_1(t), \beta_2(t)\}$. The first plot, on the left, represents $f_i(t) = \beta_0(t) + \beta_1(t)X_{i,1} + \beta_2(t)X_{i,2}$. The second plot displays $y_i(t) = f_i(t) + \epsilon_i(t)$ and the third one finally all $Y_{i,j}$. It should not surprise too much that there is a line far from the others, it happened that some of the $X_{i,j}$ was large, it is $X_{12,2} \approx -7.6$, it lays between 3σ and 4σ from the $mean(X_{i,j}) = 0$.

- **Figure 3.** Initial parameters $\{\beta_j(t)\}_{j=0,1,2}$ and a realization of the error function $\epsilon_i(t)$.

- **Figure 4.** The first two plots starting from the top left corner show the cross validation error as a function of λ. It can be seen it reaches a minimum at $\lambda \approx 0.18$. The remaining plots compare the Least Squares versus the Lasso estimation of all parameters $\beta_j(t)$. Least Squares estimations are drawn in dashed red lines, Lasso in thick black. It is evident how lasso shrunk to zero all the spurious parameters $\beta_3(t) \ldots \beta_6(t)$.

- **Figure 5.** Effect of the penalization term λ on the size and shape of functional lasso parameters $\bar{\beta}_j(t)$. Here λ takes values in the arithmetic sequence from 0 to 0.02 with 0.002 step. As we can see the effective parameters $\{\beta_0, \beta_1, \beta_2\}$ are far less sensitive to λ changes respect to the spurious ones $\{\beta_3, \beta_4, \beta_5, \beta_6\}$. This can be considered a valuable explorative tool when it is unsure if a parameter should be discarded looking only at its magnitude.

- **Figure 6.** On the left, we draw the original $\beta_j(t)$ parameters. On the right, their estimation $\hat{\beta}_j(t)$ on standardized data.

- **Figure 7.** Cross validation error for the data set with 3 active regressors. The error is monotone increasing and has minimum in $\lambda = 0$.

- **Figure 8.** Acceleration and speed curves for car B, the impacted car. Plot (a) for accelerations, (b) for velocities and (c) for standardized and smoothed velocities.

- **Figure 9.** Cross validation "in" and "out" of the sample curves. Black curves are used to estimate model parameters, red curves as cross validation test. Attention, the outlier is "in".

- **Figure 10.** Cross validation error for FL including variableΔ $widS$. There is a minimum at $\lambda \approx 0.82$ but the minimum CV value is not so different respect to CV at $\lambda \to \infty$.

- **Figure 11.** Parameters of car speed problem including varΔ $widS$ estimated by LS and FL. FL (green curves) annihilates all regressors estimated by LS (red curves) exceptedΔ $widS$ that is severely shrunk.

- **Figure 12.** Cross validation error for car accident problem, functional lasso model excludingΔ $widS$ regressor.

- **Figure 13.** Estimated parameters for car accident problem without regressorΔ $widS$. Red dashed curves are parameters estimated by Least Squares, green ones are estimated with functional Lasso.

- **Figure 14.** Parameters shape and size changes increasing λ for the car accident problem excluding variable width. Color codes are the same used in previous plots.