Adaptive sparse group LASSO in quantile regression

Álvaro Méndez Civieta[*] [†] M. Carmen Aguilera-Morillo[‡] [†]

Rosa E. Lillo[*] [†]

Abstract

This paper studies the introduction of sparse group LASSO (SGL) to the quantile regression framework. Additionally, a more flexible version, an adaptive SGL is proposed based on the adaptive idea, this is, the usage of adaptive weights in the penalization. Adaptive estimators are usually focused on the study of the oracle property under asymptotic and double asymptotic frameworks. A key step on the demonstration of this property is to consider adaptive weights based on a initial $\sqrt{n}$-consistent estimator. In practice this implies the usage of a non penalized estimator that limits the adaptive solutions to low dimensional scenarios. In this work, several solutions, based on dimension reduction techniques PCA and PLS, are studied for the calculation of these weights in high dimensional frameworks. The benefits of this proposal are studied both in synthetic and real datasets.

keywords: high-dimension; penalization; regularization; prediction; weight calculation.

[*] Department of Statistics, University Carlos III of Madrid.

[†] uc3m-Santander Big Data Institute.

[‡] Department of Applied Statistics and Operational Research, and Quality, Universitat Politècnica de València

1 Introduction

Along years, regression has become a key method in statistics. Least squares (LS) regression estimates the conditional mean response of a variable as a function of the covariates. Usually, these models assume the errors to be centered, homoscedastic and independent. Making this assumptions, it is guaranteed that the LS estimator is the best linear unbiased estimator, or a BLUE estimator. Additionally, if the errors are assumed to be Gaussian one can perform finite sample studies. However, these hypothesis are not always verified in practical applications, and the LS estimator is known to be extremely sensitive to the presence of outliers or heavy tailed distributions, making it perform poorly when the errors are non Gaussian. Ever since the seminal work of Koenker and Bassett (1978), quantile regression (QR) models have gained importance when dealing with this kind of situations. QR models allow for a relaxation of the classical first two moment conditions over the model error. In addition, the errors in QR are not required to be Gausian. This means that QR offers robust estimators capable of dealing with heteroscedasticity and outliers. QR models can also estimate different quantile levels of a response variable, giving a precise insight of the relation between response and covariates at upper and lower tails. This can provide a much richer point of view than OLS regression. For a full review on quantile regression, we recommend (Koenker, 2005).

In recent years, high dimensional data in which the number of covariates p is larger than the number of observations n ($p \gg n$), has become increasingly common. This problem can be found in many different areas like computer vision and pattern recognition (Wright et al., 2010), climate data over different land regions (Chatterjee et al., 2011), and prediction of cancer recurrence based on patients genetic information (Simon et al., 2013), (Yahya Algamal and Hisyam Lee, 2019). In these scenarios, variable selection gains in special importance offering sparse modeling alternatives that help identifying significant covariates and enhancing prediction accuracy. One of the first and most popular sparse regularization alternatives is LASSO, which was proposed by Tibshirani (1996) and adapted to the QR framework by Li and Zhu (2008), who developed the piece-wise linear solution of this technique. LASSO is a technique that penalizes each variable individually, enhancing thus individual sparsity. However, in many real applications variables are structured into groups, and group sparsity rather than individual sparsity is desired. One can think for example of a genetic dataset grouped into gene pathways. This problem was faced by the group LASSO penalization of Yuan and Lin (2006), and opened the doors to more complex penalizations like the sparse group LASSO (Friedman et al., 2010), which is a linear combination of LASSO and group LASSO providing solutions that are both between and within group sparse. With the same objective in mind, Zhou and Zhu (2010) proposed a hierarchical LASSO. Other studies have worked on properties for robust estimators in regression when the number of covariates increase with sample size (see for example Huber and Ronchetti (2009)). In the same line, it is also worth mentioning the work from Loh (2017), that extends the usage of robust estimators, like those obtained using Hubert or Tuckey loss functions (among others) to high dimensional settings, introducing a set of generalized M-estimators capable of dealing with outliers in both the errors and the covariates terms. To the best of our knowledge, the SGL technique has not been studied in the framework of QR models, so this gap is addressed first, extending the SGL penalization to quantile regression.

Zou (2006) was the first to propose the usage of adaptive weights for each variable on the LASSO penalization as a way to increase the model flexibility and correct the estimator bias. This idea, generally known as the adaptive idea, was then extended to other penalizations. The weights of the adaptive idea are defined in the literature based on an initial $\sqrt{n}$-consistent estimator. Typically, this is the result of a nonpenalized model. This definition is a key step for the demonstration of the oracle property of the estimators (in the sense of Fan and Li (2001)), but it is also restrictive, as it limits the usage of adaptive penalizations just to the situations in which solving a nonpenalized model is a feasible first step. This approach, focused on the oracle property under asymptotic, or even double asymptotic frameworks is observed in Nardi and Rinaldo (2008) for the adaptive group LASSO, Ghosh (2011) for an adaptive elastic net, Ciuperca (2019) for the adaptive group LASSO in QR, Ciuperca (2017) for the adaptive fused LASSO in QR, Wu and Liu (2009) for the adaptive LASSO and SCAD penalizations in QR, and Zhao et al. (2014) for an adaptive hierarchical LASSO in QR among others. It is especially interesting to remark the work developed by Poignard (2018), in which an adaptive sparse group LASSO estimator suitable for low dimensional scenarios (with $n > p$) is proposed, studying its theoretical properties for a set of general convex loss functions.

The main contribution of this work lies here. An adaptive sparse group LASSO (ASGL) for quantile regression estimator is defined, working especially on enabling the usage of the ASGL estimator in high dimensional scenarios (with $p \gg n$). In order to achieve this objective, four alternatives for the weight calculation step are proposed. It is worth noting that these weight calculation alternatives can be used not only in the case of the ASGL estimator, but also in the rest of the adaptive-based estimators available in the literature. The performance of these alternatives is also studied in the case of low dimensional scenarios, making the proposed work a good alternative for both high dimensional and low dimensional problems.

The rest of the paper is organized as follows. In Section 2 some basic theoretical concepts are introduced, along with the formal definition of the sparse group LASSO in quantile regression. This definition is extended to the adaptive idea in Section 3, proposing the ASGL estimator. Section 4 discusses the main results regarding asymptotic behavior of adaptive estimators, and Section 5 introduces the weights calculation alternatives for high dimensional scenarios, as well as some remarks regarding the asymptotic behavior of the proposed alternatives. Simulation results are divided into two blocks: Section 6 shows the advantages of this proposal in synthetic datasets in high and low dimensional scenarios considering a symmetric error distribution while the supplementary material shows a sensitivity analysis of the proposed methods under skewed distribution errors as well as the effect of different hyperparameter values. In Section 7 the proposed model is used in a real dataset, a genomic dataset including gene expression data of rat eye disease first shown in Scheetz et al. (2006). The computational aspects of the problem are briefly commented in Section 8, and the conclusions are provided in Section 9.

2 Penalized quantile regression

Consider a sample of n observations structured as $\mathbb{D} = (y_i, \boldsymbol{x}_i),\ i = 1, \ldots, n$ from some unknown population and define the following linear model,

$$y_i = \boldsymbol{x}_i^t \boldsymbol{\beta} + \varepsilon_i,\ i = 1, \ldots, n \tag{1}$$

where y_i is the i-*th* observation of the response variable, $\boldsymbol{x}_i \equiv (x_{i1}, \ldots, x_{ip})$ is the vector of p covariates for observation i and ε_i is the error term.

Let us introduce now the quantile regression framework by defining the loss check function,

$$\rho_\tau(u) = u(\tau - I(u < 0)) \tag{2}$$

where $I(\cdot)$ is the indicator function. In their seminal work Koenker and Bassett (1978) proved that the τ-*th* quantile of the response variable can be estimated by solving the following optimization problem,

$$\tilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ R(\boldsymbol{\beta}) \right\}. \tag{3}$$

where $R(\boldsymbol{\beta})$ defines the risk function of quantile regression,

$$R(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{x}_i^t \boldsymbol{\beta}) \tag{4}$$

Quantile regression models allow for a relaxation of the classical first two moment conditions over the model errors ε_i defined in equation 1. These errors are no longer required to be centered, homoscedastic or normally distributed, as stated in Koenker (2005), offering robust estimators capable of dealing with heteroscedasticity and outliers.

We call high dimensional scenarios to the datasets in which p is much larger than n ($p \gg n$). This problem is becoming more and more common nowadays, and can be observed in many different fields of research such as computer vision and pattern recognition (Wright et al., 2010), climate data over different land regions (Chatterjee et al., 2011) or prediction of cancer recurrence based on patients genetic information (Simon et al., 2013). An alternative that has been intensively studied in recent years for dealing with these scenarios is the penalization approach. By penalizing a regression model it is possible to perform variable selection and improve the accuracy and interpretability of the models.

One of the best known variable selection penalization methods is the least absolute selection and shrinkage operator, generally known as LASSO, proposed initially by Tibshirani (1996) which, in the case of the QR framework solves,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left\{ R(\boldsymbol{\beta}) + \lambda \left\|\boldsymbol{\beta}\right\|_1 \right\}, \tag{5}$$

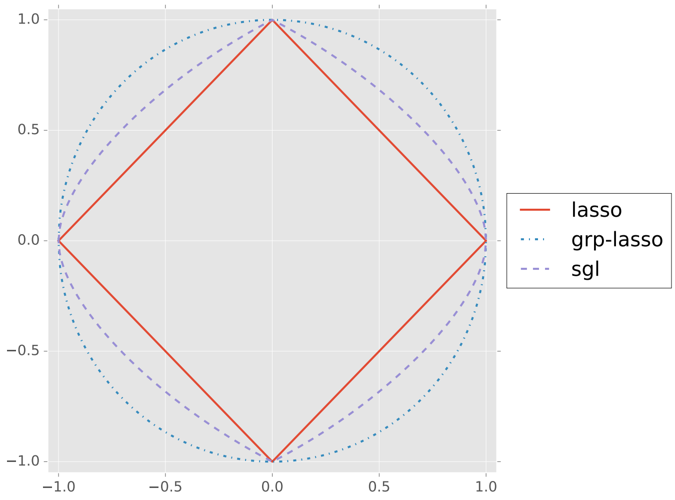
where $R(\boldsymbol{\beta})$ is the QR risk function defined in equation (4). The LASSO penalization sends many $\boldsymbol{\beta}$ components to zero, offering sparse solutions and performing automatic variable selection. In the last years, many LASSO-based algorithms have been proposed. Yuan and Lin (2006) introduced the group LASSO penalization as an answer for the need to select variables not individually but at the group level. This penalization solves the following problem,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left\{ R(\boldsymbol{\beta}) + \lambda \sum_{l=1}^{K} \sqrt{p_l} \left\|\boldsymbol{\beta}^l\right\|_2 \right\}, \tag{6}$$

where K is the number of groups, $\boldsymbol{\beta}^l \in \mathbb{R}^{p_l}$ are vectors of components of $\boldsymbol{\beta}$ from the l-*th* group, and p_l is the size of the l-*th* group. The group LASSO penalization works in a similar way to LASSO, but while LASSO enhances sparsity at individual level, group LASSO enhances sparsity at group level, selecting, or sending to zero whole groups of variables.

Initially proposed by Friedman et al. (2010), the sparse group LASSO (SGL) is a linear combination of LASSO and group LASSO penalizations. Well known in linear regression and other GLM models, to the best of our

Figure 1: Contour lines for LASSO, group-LASSO and sparse-group-LASSO penalties in the case of a single 2-dimensional group



knowledge SGL has not been adapted to QR, and as a first step in the paper, this penalization is introduced.

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{R(\boldsymbol{\beta}) + \alpha\lambda\left\|\boldsymbol{\beta}\right\|_1 + (1-\alpha)\lambda\sum_{l=1}^{K}\sqrt{p_l}\left\|\boldsymbol{\beta}^l\right\|_2\right\}. \tag{7}$$

As in LASSO and group LASSO, SGL solutions are, in general, sparse, sending many of the predictor coefficients to zero. However, while LASSO solutions are sparse at individual level, and group LASSO solutions are sparse at group level, SGL offers both between and within group sparsity, outperforming both alternatives.

From an optimization perspective, equation (7) defines a sum of convex functions. This convexity ensures that the solution of the minimization problem is a global minimum. Figure 1 shows the constrains defined by LASSO, group LASSO and SGL in the case of a single 2-dimensional group of predictors.

3 Adaptive sparse group LASSO

From an empirical perspective, sparse group LASSO shows great performance. However, due to its mathematical formulation, it applies a constant penalization rate that provides biased estimates for large coefficients. The adaptive idea, initially introduced by Zou (2006) is considered here as a way to correct this limitation. In this work, a variant of the SGL penalization, the adaptive sparse group LASSO (ASGL) for quantile regression is defined. The ASGL estimator for QR is the result of the following minimization process,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left\{ R(\boldsymbol{\beta}) + \alpha\lambda \sum_{j=1}^{p} \tilde{w}_j |\beta_j| + (1-\alpha)\lambda \sum_{l=1}^{K} \sqrt{p_l}\tilde{v}_l \left\|\boldsymbol{\beta}^l\right\|_2 \right\}, \qquad (8)$$

where $\tilde{\boldsymbol{w}} \in \mathbb{R}^p$ and $\tilde{\boldsymbol{v}} \in \mathbb{R}^K$ are known weights vectors and $R(\boldsymbol{\beta})$ is the risk function for quantile regression defined in equation 4. The intuition behind these weights is that if a variable (or group of variables) is important, it should have a small weight, and this way would be lightly penalized. On the other hand, if it is not important, by setting a large weight it is heavily penalized. This enhances the model flexibility and improves variable selection and prediction accuracy. It is worth saying that this formulation defines a convex function and thus, the global minimum can be found.

4 The oracle property

An estimator is oracle if it can correctly select the nonzero coefficients in a model with probability converging to one, and if the nonzero coefficients are asymptotically normally distributed. These properties were initially defined in Fan and Li (2001), where they proved that the SCAD was an oracle estimator under an asymptotic framework of fixed dimension p. The oracle property of the SCAD estimator was then extended in Fan and Peng (2004) to a double asymptotic framework of p depending on n. This is, $p \to \infty$ as $n \to \infty$, but p growing at a lower rate and always $n > p$. Zou (2006) proved that the LASSO was not an oracle estimator due to the bias generated by the constant penalization rate. They proposed the usage of adaptive weights as a means to correct the bias, showing that the adaptive LASSO was an oracle estimator under the asymptotic framework of fixed p, as long as the weights required by the adaptive idea were computed based on a initial $\sqrt{n}$-consistent estimator. Actually, they proposed using the result from a non penalized model for the computation of the weights $\tilde{\boldsymbol{w}}$,

$$\tilde{w_i} = \frac{1}{|\tilde{\beta}_i|^{\gamma}}, \tag{9}$$

where w_i and $\tilde{\beta}_i$ correspond to the i-*th* element of vectors $\tilde{\boldsymbol{w}}$ and $\tilde{\boldsymbol{\beta}}$ respectively, $|\cdot|$ denotes the absolute value function, γ is a non negative constant and $\tilde{\boldsymbol{\beta}}$ is the solution vector obtained from the unpenalized model (described, in the case of the QR framework, in equation (3)).

Ever since then, the adaptive idea has been extended to many LASSO-based formulations in OLS, GLM and QR models among others. One can see for instance (Ghosh, 2011) where an adaptive elastic net is defined, (Wu and Liu, 2009) that introduces the adaptive LASSO in QR, (Ciuperca, 2017) where an adaptive fused LASSO in QR is defined, (Zhao et al., 2014) who proposes an adaptive hierarchical LASSO in QR or (Poignard, 2018), where an adaptive sparse group LASSO estimator is defined in a general set of convex functions, among others. All these works are centered on the demonstration of the oracle property under the asymptotic or double asymptotic framework, being the usage of an initial $\sqrt{n}$-consistent estimator on the calculations of the weights a key step in the demonstration. A major drawback of this approach in our opinion is precisely that the asymptotic or double asymptotic frameworks are limited to low dimensional scenarios where $n > p$ but do not consider high dimensional scenarios where $p \gg n$. This is remarked by the fact that usually, the initial $\sqrt{n}$-consistent estimators used in the weight calculations are taken from non penalized models, only feasible in low dimensional scenarios.

Dealing with the problem of an increasing number of covariates is, however, challenging. When an OLS model is considered, the third order term of the taylor expansion on the loss function vanishes, but out of this framework, for example in GLM or QR models, this term does not vanish, and additional boundaries on the convergence rates of p (the number of variables) and n (the number of observations) are required in order to demonstrate the consistency and the oracle property of the estimators. This is pointed out in detail, for a general framework of convex functions, in Poignard (2018).

When considering a high dimensional scenario it is possible to find very interesting results from recent years. One can see for example (Huang et al., 2008a), who considers the oracle property of a bridge penalized least squares model under the $p \gg n$ framework as long as the bridge parameter is strictly between 0 and 1 (leaving out of the formulation the LASSO estimator). In order to achieve these results, they require additional conditions on the design matrix X, namely, they require partial orthogonality between the set of significant variables and the set of non significant variables. Similar results can be observed for the adaptive LASSO in least squares (Huang et al., 2008b) where partial orthogonality conditions are required to demonstrate the oracle property in high dimensions, for the SCAD penalization in linear models in Kim et al. (2008) and for the SCAD and MCP penalizations in quantile regression in Wang et al. (2012). However, the conditions required on the design matrix (and therefore on the covariates) to fit the oracle property are difficult to verify in practice. Thus, the results have an important mathematical relevance that should be landed in more realistic hypotheses.

5 Adaptive weights calculation

The objective of this section is to introduce different alternatives for the calculation of weights in the adaptive framework. The intuitive idea is to find a way to substitute $\tilde{\beta}$, the solution from the unpenalized model, unfeasible in high dimensional scenarios, in the calculation of the adaptive weights. This problem will be faced making use of two dimensionality reduction techniques, principal component analysis (PCA) and partial least squares (PLS). The proposed weight calculation alternatives can be used both in high dimensional and low dimensional scenarios. It is worth highlighting that these alternatives can be applied not only to the ASGL algorithm, but also to other adaptive based algorithms.

5.1 Principal components analysis

Given the covariates matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ defined in equation (1), with maximum rank $r = \min\{n, p\}$, consider the matrix of principal components $\boldsymbol{Q} \in \mathbb{R}^{p \times r}$ defined in a way such that the first principal component has the largest possible variance, and each succeeding component has the largest possible variance under the constraint that it is orthogonal to the preceding components. From an algebra perspective, the principal components in $\boldsymbol{Q}$ define an orthogonal change of basis matrix that maximize the variance explained from $\boldsymbol{X}$. Consider $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Q} \in \mathbb{R}^{n \times r}$ the projection of $\boldsymbol{X}$ into the principal components subspace. Two weight calculation alternatives based on principal components are proposed.

5.1.1 Based on a subset of components

Consider the submatrix $\boldsymbol{Q}_d = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_d]^t$ where $\boldsymbol{q}_i \in \mathbb{R}^p$ is the i-*th* column of the matrix $\boldsymbol{Q}$, and $d \in \{1, \dots, r\}$ is the number of components chosen. Let $\alpha_{pca,d} \in [0, 100]$ be the percentage of variability from $\boldsymbol{X}$ that the principal components in $\boldsymbol{Q}_d$ are able to explain. If $d = r$ then the principal components in $\boldsymbol{Q}_d$ are able to explain all the original variability from $\boldsymbol{X}$, and $\alpha_{pca,d} = 100$. If $d < r$ then $\alpha_{pca,d} < 100$. The number of components chosen in order to explain up to a certain percentage of variability is fixed by the researcher. Obtain $\boldsymbol{Z}_d = \boldsymbol{X}\boldsymbol{Q}_d \in \mathbb{R}^{n \times d}$ the projection of $\boldsymbol{X}$ into the subspace generated by $\boldsymbol{Q}_d$ and solve the unpenalized model,

$$\tilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^d}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{z}_i^t \boldsymbol{\beta}) \right\}. \tag{10}$$

This model defines a low dimensional scenario where $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^d$. Using this solution, it is possible to obtain an estimation of the high dimensional scenario solution, $\hat{\boldsymbol{\beta}} = \boldsymbol{Q}_d \tilde{\boldsymbol{\beta}} \in \mathbb{R}^p$. Finally, the weights are estimated as,

$$\tilde{w_j} = \frac{1}{|\hat{\beta}_j|^{\gamma_1}} \quad \text{and} \quad \tilde{v_l} = \frac{1}{\left\| \hat{\boldsymbol{\beta}}^l \right\|_2^{\gamma_2}}, \tag{11}$$

where $\hat{\beta}_j$ is the j-*th* component from $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}^l$ is the vector of components of $\boldsymbol{\beta}$ from the l-*th* group, and γ_1 and γ_2 are non negative constants usually taken in $[0, 2]$.

5.1.2 Based on the first component

A more straightforward approach based on the first principal component is also proposed. The principal components are no more than linear combinations of the original variables. Therefore, the first principal component $\boldsymbol{q}_1 \in \mathbb{R}^p$, which is the first column of the matrix $\boldsymbol{Q}$, includes one weight for each of the p original variables. This proposal consists of calculating the weights as,

$$\tilde{w_j} = \frac{1}{|q_{1j}|^{\gamma_1}} \quad \text{and} \quad \tilde{v_l} = \frac{1}{\left\|\boldsymbol{q}_1^l\right\|_2^{\gamma_2}}, \tag{12}$$

where q_{1j} is the j-*th* component from $\boldsymbol{q}_1$ and defines the weight associated to the j-*th* original variable, $\boldsymbol{q}_1^l$ is the vector of components of $\boldsymbol{q}_1$ from the l-*th* group and γ_1 and γ_2 are non negative constants usually taken in $[0, 2]$.

5.2 Partial least squares

The principal components are defined in a way such that they capture the maximum possible variance from $\boldsymbol{X}$ under the constraint that they are orthogonal to the rest of the principal components. However, being relevant for describing the variance of $\boldsymbol{X}$ does not necessarily mean that a principal component is relevant for predicting the value of $\boldsymbol{y}$. Partial least squares (PLS) is a dimensionality reduction technique centered on maximizing the covariance between $\boldsymbol{X}$ and $\boldsymbol{y}$.

Given the covariates matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ defined in equation (1), with maximum rank $r = \min\{n, p\}$, consider the matrix of PLS components $\boldsymbol{T} \in \mathbb{R}^{p \times s}$ and the projection of $\boldsymbol{X}$ into the subspace generated by $\boldsymbol{T}$: $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{T} \in \mathbb{R}^{n \times s}$.

The matrix of PLS components $\boldsymbol{T}$ defines a nonorthogonal change of basis matrix whose projection $\boldsymbol{U}$ is computed in a way such that the first projection vector, $\boldsymbol{u_1} \in \mathbb{R}^n$ has the largest possible covariance with $\boldsymbol{y}$, and each succeeding projection vector has the largest possible covariance with $\boldsymbol{y}$ under the constraint that it is uncorrelated to the rest of the projection vectors.

Given the submatrix $\boldsymbol{T}_d = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_d]^t$ where $\boldsymbol{t}_i \in \mathbb{R}^p$ is the i-*th* column of the matrix $\boldsymbol{T}$, and $d \in \{1, \ldots, s\}$ is the number of components chosen, let $\alpha_{pls,d} \in [0, 100]$ be the percentage of variability from $\boldsymbol{X}$ that the PLS components in $\boldsymbol{T}_d$ are able to explain. The nonorthogonality of $\boldsymbol{T}$ implies that the total number of PLS components available to be computed is smaller than the rank of $\boldsymbol{X}$, $s \leq r$, and that the maximum possible percentage of variability explained by the PLS components $\alpha_{pls,s}$ is then lower than 100%.

In the case of principal components analysis, the matrix of principal components $\boldsymbol{Q}$ defines an orthogonal change of basis matrix that results into an orthogonal projection matrix $\boldsymbol{Z}$ maximizing the variance of $\boldsymbol{X}$. On the other hand, PLS defines a nonnecesarily orthogonal change of basis matrix $\boldsymbol{T}$ that results into an uncorrelated projection matrix $\boldsymbol{U}$ maximizing the covariance between $\boldsymbol{U}$ and $\boldsymbol{y}$. In the same way as for the PCA alternatives proposed, two alternatives of weight calculation using PLS are considered: based on a subset of PLS components, and based just on the first PLS component.

5.3 Influence of PCA and PLS on the oracle property

As commented in Section 4, a key condition in the demonstration of the oracle property in adaptive estimators is to assume that the initial estimator used in the weights calculation is $\sqrt{n}$-consistent.

The usage of pca_d or pls_d weight calculation proposes to consider a subset of d components in the estimation of the weights. A question that may arise here is whether these PCA (or PLS) estimator is $\sqrt{n}$-consistent or not. We propose the following simple low dimensional example in the OLS framework that can help answering this question.

Example:

Given the random variables $X_1 \sim N(0, 0.99)$ and $X_2 \sim N(0, 0.01)$, consider the random vector : $X = (X_1, X_2)$, for which

$$cov(X) = \begin{pmatrix} 0.99 & 0 \\ 0 & 0.01 \end{pmatrix}.$$

And thus, the eigenvalues from cov(X) are $\lambda_1 = 0.99$ and $\lambda_2 = 0.01$, and the matrix of eigenvectors is

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

If PCA is applied on this random vector X, the rotation matrix obtained will be P, yielding to a first principal component that explains 99% of the original variability and a second principal component that explains the remaining 1%.

Consider now the following linear model,

$$y = X\beta + \varepsilon,$$

where $\beta = (0, 100)^t$ and $\varepsilon \sim N(0, 0)$. Following the steps described in Section 5.1.1, consider a subset of components that explain up to a certain percentage of variability, for example, 99% of the variability. This implies that X will be projected onto the subspace spanned just by the first principal component P_1,

$Z = XP_1 = X_1$. Solve now the linear model $\tilde{y} = Z\tilde{\beta}$, where

$$\tilde{\beta} = \frac{cov(Z, y)}{var(Z)} = \frac{cov(X_1, y)}{var(X_1)} = 0.$$

Then, the projection of the estimator $\tilde{\beta}$ into the original subspace is given by $\hat{\beta} = P_1\tilde{\beta} = (0, 0)^t$. Now, in order to be $\sqrt{n}$-consistent, an estimator should verify:

$$(\hat{\beta} - \beta) \text{ is } O_p(n^{-1/2}) \text{ if for all } \varepsilon > 0 \; \exists K > 0 \text{ s.t.}$$

$$P_{n\to\infty}(\sqrt{n}|\hat{\beta} - \beta| > K) < \varepsilon$$

Taking into account that $\beta = (0, 100)^t$, it is clear that the $\sqrt{n}$-consistency property is not verified by $\hat{\beta}$. The problem arises because the variability in variable Y is explained by X_2, which is not selected because it explains only 1% of the total variability of X.

We would like to point out that this example is meant to be a counterexample of a situation in which the pca_d is not $\sqrt{n}$-consistent. However, in our opinion, it clarifies the conditions required by the estimator in order to be consistent, as stated in the following remarks.

Remark 1. Consider an ASGL estimator, where the weights are computed based on a subset of principal components pca_d in the asymptotic or double asymptotic frameworks. If all the components are selected (this is, if the components explain 100% of the original variability), then the initial estimator used in the weights calculation is $\sqrt{n}$-consistent, and therefore, the ASGL estimator is an oracle estimator. Observe that by selecting all the components, $\hat{\beta} = Q\tilde{\beta}$ is equal to the unpenalized estimator defined in equation (3).

Remark 2. As shown in Section 4, the proof of the oracle property of an estimator in high dimensional scenarios is much more complex than in low dimensional scenarios. We conjecture that in the high dimensional context, the pca_d estimator will behave in a similar way as in low dimensional scenarios, requiring to achieve a 100% of explained variability, but requiring also additional hypothesis similar to the ones observed in, for example, Wang et al. (2012). In this paper, a set of 5 previous conditions is required for the demonstration of the oracle property in a high dimensional framework in quantile regression while considering non convex penalizations (such as SCAD). Among other things, the proposed conditions include restrictions on the design matrix, for example, that given the design matrix $\boldsymbol{X}$, $\boldsymbol{S} = \frac{1}{n}\boldsymbol{X^t X}$ should be bounded, and the eigenvalues of $\boldsymbol{S}$ should be bounded as well. We consider that due to the complexity of the required results, studying the theoretical aspect of the estimator in high dimensional scenarios is a topic for further work. However, we study the behavior of this estimator in high dimensional scenarios both in synthetic and real datasets in Sections 6 and 7, and in the supplementary material, obtaining very good results.

Remark 3. The study of the oracle property of the pls_d estimator is much more complex than this of pca_d. As commented in section 5.2, the maximum percentage of variability explained by the PLS components can be smaller than 100%, and thus, we would be facing the same issues described in the example above. This situation will also be a topic for further work.

6 Simulation study: symmetric errors

This section shows the performance of the proposed ASGL estimator under different synthetic dataset examples focused on symmetric errors as it is usual in OLS models. The proposed ASGL estimator is studied here under the framework of the following model,

$$y = X\beta + \varepsilon, \; \varepsilon \sim t(3),$$

where the data matrix X is generated from a standard Gaussian distribution. Variables are organized in groups, considering a within group correlation of 0.5 and a between group correlation of 0. A quantile level $\tau = 0.5$ is considered. The scheme used here is an adaptation of other simulation schemes used in Wu and Liu (2009) and Zhao et al. (2014).

Given that the ASGL formulation in equation (8) includes a weight penalization on the group LASSO part based on the group size (the term $\sqrt{p_l}$), two model formulations are considered:

- Adaptive LASSO in sparse group LASSO (AL-SGL), where $\tilde{\boldsymbol{w}} \neq \mathbf{1}$ but $\tilde{\boldsymbol{v}} = \mathbf{1}$, in which the adaptive idea is only applied to the LASSO part.
- Adaptive sparse group LASSO (ASGL), where $\tilde{\boldsymbol{w}} \neq \mathbf{1}$ and $\tilde{\boldsymbol{v}} \neq \mathbf{1}$.

Furthermore, the four weight calculation alternatives proposed are studied:

- PCA weights based on regression on a subset of principal components, we denote this as pca_d;
- PCA weights based on the first principal component, we denote this as pca_1;

- PLS weights based on regression on a subset of PLS components, we denote this as pls_d;
- PLS weights based on the first PLS component, we denote this as pls_1.

The total number of components d used in the weight estimation in pls_d and pca_d is chosen such that in both cases the percentage of variability explained from the original matrix $\boldsymbol{X}$ is $\alpha_{pca,d} = 80\%$, $\alpha_{pls,d} = 80\%$. As commented along Section 5, due to the non orthogonality of the PLS components it can happen that the maximum possible variability explained by the PLS components $\alpha_{pls,s}$ is smaller than 80%. In these cases we consider d such that $\alpha_{pls,d} = \alpha_{pls,s}$.

The results obtained by the models proposed in this work are compared with the results from LASSO and SGL formulations. For each dataset $\mathbb{D}$, a partition into three disjoint subsets, $\mathbb{D}_{train}$, $\mathbb{D}_{val}$ and $\mathbb{D}_{test}$ is considered. $\mathbb{D}_{train}$ is used for training the models, this is, solving the model equations. $\mathbb{D}_{val}$ is used for validation, this is, optimizing the model parameters. This optimization is performed based on grid-search. Finally, $\mathbb{D}_{test}$ is used for testing the models prediction accuracy. The model parameters are optimized based on the minimization of the quantile error, defined as,

$$E_v = \frac{1}{\#\mathbb{D}_{val}} \sum_{(y_i, \boldsymbol{x_i}) \in \mathbb{D}_{val}} \rho_\tau(y_i - \boldsymbol{x}_i^t \hat{\boldsymbol{\beta}}), \tag{13}$$

where $\rho_\tau(\cdot)$ denotes the quantile function defined at (2), and # denotes the cardinal of a set. The final model error is calculated over $\mathbb{D}_{test}$ as,

$$E_t = \frac{1}{\#\mathbb{D}_{test}} \sum_{(y_i, \boldsymbol{x_i}) \in \mathbb{D}_{test}} \rho_\tau(y_i - \boldsymbol{x}_i^t \hat{\boldsymbol{\beta}}). \tag{14}$$

Additionally, the following metrics evaluating the performance of the methods are considered:

- $\left\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\|_2$ the euclidean distance between the estimated vector and the true vector;

- true positive rate (TPR)= $\mathrm{P}(\hat{\beta}_i \neq 0|\beta_i \neq 0)$;

- true negative rate (TNR)= $\mathrm{P}(\hat{\beta}_i = 0|\beta_i = 0)$;

- correct selection rate (CSR)= $\mathrm{P}(\hat{\beta} = \beta)$.

We are interested in studying the performance of the proposed models under different situations. An aspect to be analysed is the effect of an increase on the number of variables, and regarding this aspect, three cases will be considered:

- high-dimensional case with 625 variables;

- high-dimensional case with 225 variables;

- low dimensional case with 100 variables.

Additionally, another important factor is the spread of the significant variables among different groups. In order to study this aspect, two cases will be considered:

- sparse distribution of significant variables: significant variables are spread among many groups, but there is no group fully formed by significant variables;

- dense distribution of significant variables: significant variables are concentrated into a few number of groups, fully formed by significant variables.

Varying the number and the spread of the variables, six cases will be studied:

Case 1: sparse distribution of 625 *variables*

There are 25 groups of size 25 each, a total number of 625 variables. Among these groups, 7 groups with 8 significant variables each are defined, a total number of 56 significant variables. For $l \in \{1 \dots, 25\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1, 2, \dots, 8, \underbrace{0, \dots, 0}_{17}), \ l = 1, \dots, 7 \\ \beta^l &= (\underbrace{0, \dots, 0}_{25}), \ l = 8, \dots, 25. \end{cases}$$

Case 2: dense distribution of 625 *variables*

There are 25 groups of size 25 each, a total number of 625 variables. Among these groups, 3 groups with 25 significant variables each are defined, a total number of 75 significant variables. For $l \in \{1 \dots, 25\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1, 2, \dots, 25), \ l = 1, \dots, 3 \\ \beta^l &= (\underbrace{0, \dots, 0}_{25}), \ l = 4, \dots, 25. \end{cases}$$

Case 3: sparse distribution of 225 *variables*

There are 15 groups of size 15 each, a total number of 225 variables. Among these groups, 7 groups with 8 significant variables each are defined, a total number of 56 significant variables. For $l \in \{1 \ldots, 15\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1, 2, \ldots, 8, \underbrace{0, \ldots, 0}_{7}), \ l = 1, \ldots, 7 \\ \beta^l &= (\underbrace{0, \ldots, 0}_{15}), \ l = 8, \ldots, 15. \end{cases}$$

Case 4: dense distribution of 225 *variables*

There are 15 groups of size 15 each, a total number of 225 variables. Among these groups, 3 groups with 15 significant variables each are defined, a total number of 45 significant variables. For $l \in \{1 \ldots, 15\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1, 2, \ldots, 15), \ l = 1, \ldots, 3 \\ \beta^l &= (\underbrace{0, \ldots, 0}_{15}), \ l = 4, \ldots, 15. \end{cases}$$

Case 5: sparse distribution of 100 *variables*

There are 10 groups of size 10 each, a total number of 100 variables. Among these groups, 5 groups with 6 significant variables each are defined, a total number of 30 significant variables. For $l \in \{1 \ldots, 10\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1,2,\dots,6,\underbrace{0,\dots,0}_{4}),\ l=1,\dots,5 \\ \beta^l &= (\underbrace{0,\dots,0}_{10}),\ l=6,\dots,10. \end{cases}$$

Case 6: dense distribution of 100 *variables*

There are 10 groups of size 10 each, a total number of 100 variables. Among these groups, 3 groups with 10 significant variables each are defined, a total number of 30 significant variables. For $l \in \{1\dots,10\}$, coefficients inside each group are defined as,

$$\begin{cases} \beta^l &= (1,2,\dots,10),\ l=1,\dots,3 \\ \beta^l &= (\underbrace{0,\dots,0}_{10}),\ l=4,\dots,10. \end{cases}$$

We consider that *Case 1* is the most representative example in further applications, and therefore it will be intensively studied here, and also in the simulations regarding the sensitivity analysis shown in the supplementary material. Each simulation example has been executed 50 times considering 100/100/5000 observations in the train / validate / test samples, except in the low dimensional simulations (*Case* 5 and 6) where 500/500/5000 observations were considered. The large test sets formed by 5000 observations help increase the stability of the results, however, models are built using train and validate sets, making the 625 variables and 225 variables simulations high dimensional ($p > n$). The results have been summarized in terms of the mean and standard deviation values (shown in parenthesis), and the best result from each metric is highlighted.

As it was commented in Section 4, the general tendency found in the literature regarding the weights in adaptive models is to define them based on the results of the unpenalized model,

$$\tilde{w_i} = \frac{1}{|\tilde{\beta}_i|^\gamma}, \tag{15}$$

where w_i and $\tilde{\beta}_i$ correspond to the i-*th* element of vectors $\tilde{\boldsymbol{w}}$ and $\tilde{\boldsymbol{\beta}}$ respectively, $|\cdot|$ denotes the absolute value function, γ is a non negative constant and $\tilde{\boldsymbol{\beta}}$ is the solution vector obtained from the unpenalized model (described, in the case of the QR framework, in equation (3)). This approach is limited just to low dimensional scenarios, where the unpenalized model can actually be solved. For this reason, in the low dimensional cases, the results of the proposed models are compared with the results from the weights based on the unpenalized model.

6.1 Simulation 1: sparse distribution of significant variables.

This simulation shows the results obtained under simulation *Case* 1, considering 625 variables, *Case* 3, considering 225 variables and *Case* 5, considering 100 variables. In all of them, the variables are sparsely distributed among groups, and a symmetric error from a t(3) is considered.

Results from this simulation scheme are displayed in Table 1, which is divided into three parts related to the three *Cases* under study. The first part of the table analyses *Case* 1, which considers 625 variables. In this part, the results from LASSO and SGL are compared against the eight proposed weight

Table 1: Simulation 1. Sparse distribution of variables. Considering a t(3) error.

	$\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
		$p = 625$ variables			
LASSO	23.37 (4.61)	7.85 (1.70)	**0.89** (0.01)	0.76 (0.07)	**0.90** (0.01)
SGL	19.62 (3.28)	6.29 (1.08)	0.76 (0.10)	0.90 (0.04)	0.75 (0.12)
AL-SGL-pca_d	17.97 (3.56)	5.68 (1.13)	0.83 (0.07)	0.88 (0.05)	0.83 (0.08)
AL-SGL-pca_1	21.41 (2.78)	6.88 (0.93)	0.70 (0.10)	0.90 (0.04)	0.68 (0.12)
AL-SGL-pls_d	17.60 (3.28)	5.78 (1.14)	0.83 (0.06)	0.89 (0.04)	0.83 (0.07)
AL-SGL-pls_1	19.40 (2.99)	6.23 (0.99)	0.78 (0.09)	0.90 (0.04)	0.77 (0.10)
ASGL-pca_d	15.19 (3.43)	4.65 (1.04)	0.84 (0.04)	**0.92** (0.03)	0.83 (0.04)
ASGL-pca_1	21.38 (2.58)	6.80 (0.87)	0.73 (0.10)	0.91 (0.04)	0.71 (0.11)
ASGL-pls_d	**13.23** (3.35)	**4.07** (0.99)	0.85 (0.03)	0.91 (0.04)	0.84 (0.04)
ASGL-pls_1	17.56 (3.98)	5.61 (1.33)	0.81 (0.01)	0.91 (0.04)	0.80 (0.07)
ASGL-$spls_d$	14.31 (3.30)	4.36 (0.99)	0.85 (0.03)	0.92(0.04)	0.84 (0.04)
ASGL-$spca_d$	18.05 (3.19)	5.75 (1.06)	0.78 (0.07)	0.91(0.03)	0.77 (0.08)
		$p = 225$ variables			
LASSO	8.09 (2.48)	2.66 (0.81)	**0.80** (0.02)	0.96 (0.03)	**0.75** (0.02)
SGL	6.43 (2.02)	2.12 (0.60)	0.76 (0.06)	0.98 (0.02)	0.69 (0.07)
AL-SGL-pca_d	6.66 (2.33)	2.20 (0.76)	0.78 (0.06)	0.97 (0.03)	0.71 (0.08)
AL-SGL-pca_1	7.06 (1.98)	2.30 (0.61)	0.73 (0.06)	0.98 (0.02)	0.65 (0.09)
AL-SGL-pls_d	6.95 (1.79)	2.28 (0.56)	0.77 (0.06)	0.97 (0.02)	0.70 (0.08)
AL-SGL-pls_1	7.27 (2.46)	2.39 (0.78)	0.74 (0.06)	0.98 (0.02)	0.66 (0.08)
ASGL-pca_d	5.09 (1.32)	1.70 (0.38)	0.73 (0.09)	**0.99** (0.01)	0.65 (0.12)
ASGL-pca_1	7.07 (1.98)	2.31 (0.62)	0.75 (0.06)	0.98 (0.02)	0.67 (0.07)
ASGL-pls_d	**5.05** (1.30)	**1.68** (0.37)	0.74 (0.09)	**0.99** (0.02)	0.66 (0.12)
ASGL-pls_1	6.21 (1.78)	2.04 (0.52)	0.74 (0.05)	0.98 (0.02)	0.66 (0.06)
		$p = 100$ variables			
LASSO	0.59 (0.08)	0.59 (0.01)	0.79 (0.09)	**1.00** (0.00)	0.69 (0.14)
SGL	0.60 (0.08)	0.59 (0.01)	0.75 (0.11)	**1.00** (0.00)	0.64 (0.16)
ASGL-pca_d	0.55 (0.08)	0.58 (0.01)	0.81 (0.10)	**1.00** (0.00)	0.73 (0.14)
ASGL-pls_d	**0.45** (0.07)	**0.58** (0.06)	0.95 (0.07)	**1.00** (0.00)	0.93 (0.09)
ASGL-unpenalized	**0.45** (0.07)	**0.58** (0.05)	**0.96** (0.07)	**1.00** (0.00)	**0.95** (0.07)

Figure 2: Simulation 1. Sparse distribution of 625 variables. Considering a t(3) error. Box-plots showing the test error of the different models.

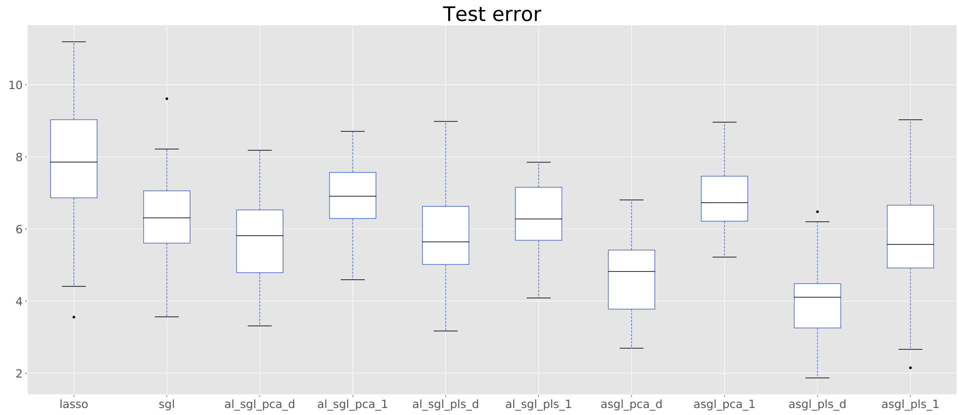
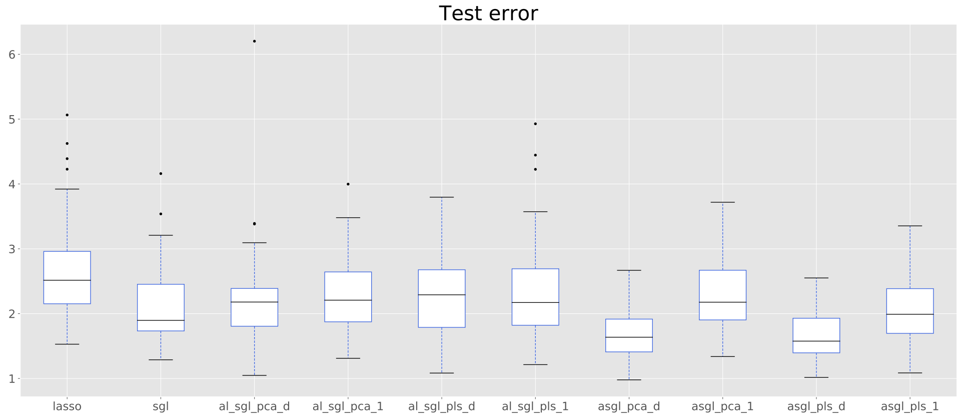


Figure 3: Simulation 1. Sparse distribution of 225 variables. Considering a t(3) error. Box-plots showing the test error of the different models.



calculation alternatives commented before. Additionally, the performance of sparse variations of PCA and PLS is studied. These alternatives appear denoted as $spca_d$ (from sparse PCA) and $spls_d$ (from sparse PLS). Sparse PCA was initially proposed by (Zou et al., 2006) as a method that computes principal components adding a LASSO based penalization to standard PCA. This yields to principal components that are sparse linear combinations of the original variables, though are no longer orthogonal. In the same sense, Chun and Keleş (2010) proposed an sparse alternative to PLS. Both alternatives are studied in this simulation.The best results here are obtained by the ASGL model using pls_d weights, closely followed by $spls_d$ and pca_d weights. This model outperforms LASSO and SGL both in terms of the distance between predicted and true $\boldsymbol{\beta}$, and in terms of the test error E_t. Given that LASSO enhances individual sparsity, LASSO solutions are more sparse than the solutions obtained by the proposed models , and this is shown in the TNR values. However, LASSO offers poor results in terms of the TPR (this is, in terms of the selection of the truly significant variables). SGL shows the opposite behavior, producing solutions with large TPR values but low TNR values. Compared to these techniques, the proposed ASGL formulations achieve good variable selection results both in terms of TNR and TPR. It is worth highlighting the results achieved using the sparse PCA ($spca_d$) and sparse PLS ($spls_d$) weights alternatives. As can be seen, the performance of $spca_d$ and $spls_d$ is worse than that of pls_d. Our guess is that establishing a double-sparsity framework, namely, sparse components used to estimate prior weights for an adaptive sparse group LASSO, is not that beneficial, and that simple PLS may be sufficient for the weight calculation, leaving the achievement of sparse solutions to the effect of the ASGL estimator. Additionally, using sparse PCA or sparse PLS in the weight calculation requires to optimize a series of parameters related to these techniques, and then another series of parameters related to the ASGL estimator. Finding the optimal solution in such a grid of parameters can be numerically cumbersome and time-consuming.

A similar behavior is observed in *Case 3*, that considers 225 variables. As before, the best results in terms of prediction accuracy are provided by ASGL pls_d and pca_d alternatives. Finally, the study performed in the low dimensional *Case* 5 is centered on the models achieving the best results among the proposals considered, namely pls_d and pca_d weights, that are compared against LASSO and SGL penalizations, and against the ASGL unpenalized, which is feasible only in this low dimensional framework and that consists in estimating the weights based on a unpenalized model (as it is usually done in the literature). It is worth to remark here that the pls_d alternative performs just as well as the *unpenalized* one, which is a nice finding of this approach.

Figures 2 and 3 display box-plots of the test error E_t for different models in the high dimensional frameworks, showing that the spread of E_t is much smaller in the ASGL pls_d and pca_d than in the LASSO and SGL, indicating that these models provide more stable solutions in terms of prediction accuracy.

6.2 Simulation 2: dense distribution of significant variables.

This simulation shows the results obtained under simulation *Case* 2, considering 625 variables, *Case* 4, considering 225 variables and *Case* 6, considering 100 variables. In all of them, the variables are densely distributed among groups, and a symmetric error from a t(3) is considered.

The results from this simulation scheme are displayed in Table 2. Similar to

Table 2: Simulation 2. Dense distribution of variables. Considering a t(3) error.

	$\left\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
		$p = 625$ variables			
LASSO	21.00 (13.00)	7.13 (4.67)	**0.95** (0.01)	0.96 (0.03)	**0.95** (0.01)
SGL	6.02 (1.77)	1.99 (0.56)	0.82 (0.09)	**1.00** (0.01)	0.80 (0.10)
AL-SGL-pca_d	4.32 (0.99)	1.45 (0.28)	0.94 (0.04)	**1.00** (0.01)	0.93 (0.05)
AL-SGL-pca_1	7.17 (2.47)	2.30 (0.75)	0.72 (0.09)	**1.00** (0.01)	0.68 (0.11)
AL-SGL-pls_d	4.81 (1.47)	1.60 (0.44)	0.92 (0.06)	**1.00** (0.01)	0.90 (0.07)
AL-SGL-pls_1	5.38 (1.20)	1.77 (0.57)	0.87 (0.08)	**1.00** (0.01)	0.85 (0.09)
ASGL-pca_d	**3.61** (0.78)	**1.23** (0.20)	0.92 (0.10)	**1.00** (0.01)	0.90 (0.12)
ASGL-pca_1	7.60 (3.20)	2.46 (1.01)	0.74 (0.09)	**1.00** (0.01)	0.71 (0.11)
ASGL-pls_d	3.85 (0.83)	1.29 (0.21)	0.85 (0.03)	**1.00** (0.01)	0.89 (0.13)
ASGL-pls_1	4.17 (1.17)	1.40 (0.32)	0.90 (0.11)	**1.00** (0.01)	0.87 (0.09)
		$p = 225$ variables			
LASSO	4.43 (1.10)	1.57 (0.35)	0.87 (0.03)	0.99 (0.01)	0.83 (0.05)
SGL	3.29 (0.75)	1.21 (0.21)	0.73 (0.13)	0.99 (0.01)	0.64 (0.17)
AL-SGL-pca_d	2.88 (0.50)	1.07 (0.14)	0.78 (0.06)	**1.00** (0.01)	0.84 (0.11)
AL-SGL-pca_1	3.63 (0.73)	1.30 (0.22)	0.61 (0.15)	0.99 (0.01)	0.47 (0.21)
AL-SGL-pls_d	2.92 (0.57)	1.09 (0.16)	0.84 (0.12)	**1.00** (0.01)	0.78 (0.16)
AL-SGL-pls_1	3.14 (0.65)	1.16 (0.18)	0.76 (0.14)	**1.00** (0.01)	0.67 (0.20)
ASGL-pca_d	**2.56** (0.49)	**0.98** (0.13)	**0.89** (0.12)	**1.00** (0.01)	**0.85** (0.16)
ASGL-pca_1	3.49 (0.79)	1.25 (0.22)	0.62 (0.15)	**1.00** (0.01)	0.49 (0.21)
ASGL-pls_d	2.59 (0.43)	0.99 (0.10)	0.88 (0.16)	**1.00** (0.01)	0.83 (0.21)
ASGL-pls_1	2.80 (0.53)	1.05 (0.14)	0.81 (0.12)	**1.00** (0.01)	0.74 (0.17)
		$p = 100$ variables			
LASSO	0.52 (0.08)	0.58 (0.01)	0.82 (0.10)	**1.00** (0.00)	0.75 (0.13)
SGL	0.50 (0.08)	0.58 (0.01)	0.74 (0.17)	**1.00** (0.00)	0.63 (0.24)
ASGL-pca_d	0.45 (0.07)	**0.57** (0.01)	0.92 (0.11)	**1.00** (0.00)	0.88 (0.15)
ASGL-pls_d	**0.44** (0.07)	**0.57** (0.01)	**0.95** (0.07)	**1.00** (0.00)	**0.93** (0.10)
ASGL-unpenalized	0.45 (0.07)	**0.57** (0.01)	0.92 (0.12)	**1.00** (0.00)	0.89 (0.17)

Figure 4: Simulation 2. Dense distribution of 625 variables. Considering a t(3) error. Box-plots showing the test error of the different models.

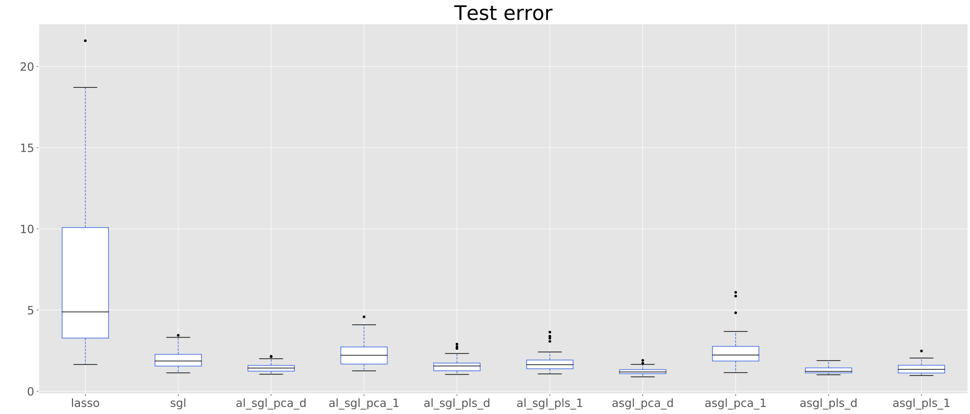
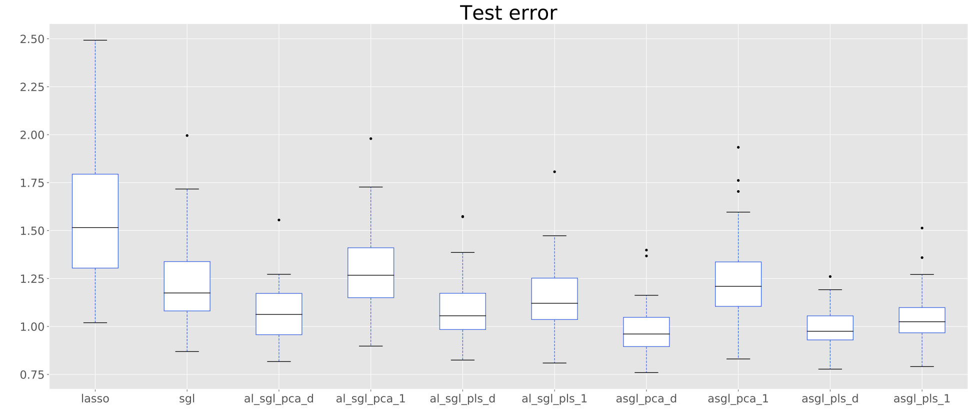


Figure 5: Simulation 2. Dense distribution of 225 variables. Considering a t(3) error. Box-plots showing the test error of the different models.



the situation shown in the sparse distribution simulation, the ASGL model using pls_d or pca_d weights shows the best results in terms of the distance between predicted and true $\boldsymbol{\beta}$, and the value of E_t in the high dimensional cases. These proposals offer also the best compromise between TPR and TNR. It is worth saying that under a more "compact" distribution of the significant variables in a small number of groups, the proposed methods show a great improvement in terms of prediction accuracy compared to LASSO and SGL. As before, the low dimensional *case* is studied centered on the models achieving the best results among the proposals considered, pls_d and pca_d weights, that are compared against LASSO, SGL and ASGL *unpenalized* penalizations. It can be seen here that pls_d is the one achieving the best results in this framework, closely followed by pca_d and *unpenalized* results.

Figures 4 and 5 display box-plots of test error value E_t in high dimensional scenarios, showing, as in the previous simulation scheme, that ASGL models with pls_d or pca_d weights also provide more stable results in terms of spread. Based on previous simulations, we conclude that the best performance both in the high dimensional and low dimensional frameworks, considering sparse or dense distribution of significant variables is achieved by ASGL models with pls_d or pca_d weights.

Additionally to the simulations shown here, a comprehensive sensitivity analysis that studies the behavior of the proposed methodology under different non symmetric error distributions, when varying the powers γ_1 and γ_2 entering the weights and when varying the number of PCA and PLS components chosen in the weight calculation can be found in the supplementary material.

7 Real application

The performance of the ASGL estimator is shown here using a genomic dataset first reported in Scheetz et al. (2006). The dataset consists of 120 twelve-week-old male offspring animals chosen for tissue harvesting from the eyes and for micro-array analysis. The dataset contains expression values from 31042 different probe-sets (Affymetric GeneChip Rat Genome 230 2.0 Array) on a logarithmic scale. As described in Huang et al. (2008b) and Wang et al. (2012), a two-steps preprocessing is performed, selecting, among the 31042 probe-sets, the ones that are sufficiently expressed, and sufficiently variable. A probe is considered to be sufficiently expressed if the maximum expression value observed for that probe among the 120 animals is greater than the 25-*th* percentile of the entire set of RMA expression values. A probe is considered to be sufficiently variable if it shows at least 2-fold variation in the expression value among the 120 rats. There are 18986 probes that meet these criteria.

We study how expression level of gene TRIM32, corresponding to probe 1389163_at, is related to expression levels at other probes. Chiang et al. (2006) pointed out that gene TRIM32 was found to cause Bardet-Biedl syndrome, a disease of multiple organ systems including the retina.(Scheetz et al., 2006, :1) stated: “Any genetic element that can be shown to alter the expression of a specific gene or gene family known to be involved in a specific disease is itself an excellent candidate for involvement in the disease, either primarily or as a genetic modifier.” Here the sample size is 120 (the number of animals selected for micro-array analysis), and the number of covariates (probes that pass the preprocessing steps) is 18985. The correlation coefficients of the 18985 probes and the probe corresponding to gene TRIM32 is calculated, and the genes in which the absolute value of the correlation exceeds 0.5 are selected. There are

3734 probes meeting this criteria. Finally, this dataset is standardized. Only a few genes are expected to be related to gene TRIM32, making this a high dimensional sparse problem.

From a biological perspective it is clear that genes do not work individually. The problem of grouping genes based on a medical criteria is nowadays under intense study, and it is possible to find some group structures for human genetic information based, for example, in cytogenetic positions (Subramanian et al., 2005). It is interesting to remark that groups built based on biological criteria are usually formed just by a few dozens of genes. For example, in the case of groups based on cytogenetic positions, groups averaged 30 genes, as stated in Simon et al. (2013). However, these group structures are not available for all the genetic information, and to the best of our knowledge there is no genetic grouping alternative for the dataset under study here.

We address the grouping problem from an statistical perspective, using principal components analysis to create groups of genes that are similar. It is worth to remark that in Section 5.1 PCA was used for estimating the ASGL weights, while here it will be used for variable clustering.

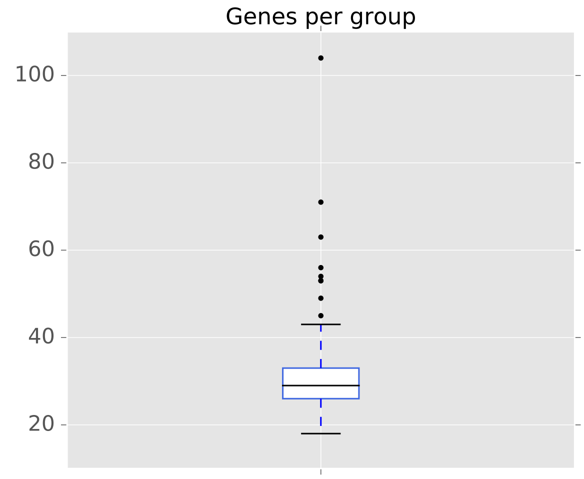
Variable clustering using PCA

1. Given a matrix of covariates $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ as in Section 5.1, obtain the matrix of principal components $\boldsymbol{Q} \in \mathbb{R}^{p \times r}$ $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ defined in Section 5.1.

2. Consider r possible groups, as many as principal components.

3. Each principal component $\boldsymbol{q}_i \in \boldsymbol{Q}$, $i \in 1, \dots, r$, is a linear combination of the original variables from $\boldsymbol{X}$. Assign each original variable to the

Figure 6: Gene expression data of rat eye disease. Box-plot showing the sizes of the groups built using PCA.



group associated to the principal component in which that variable had its maximum weight (in absolute value).

The intuition behind this process is that variables with a large weight in the same principal component are likely to be related and should be included in the same group.

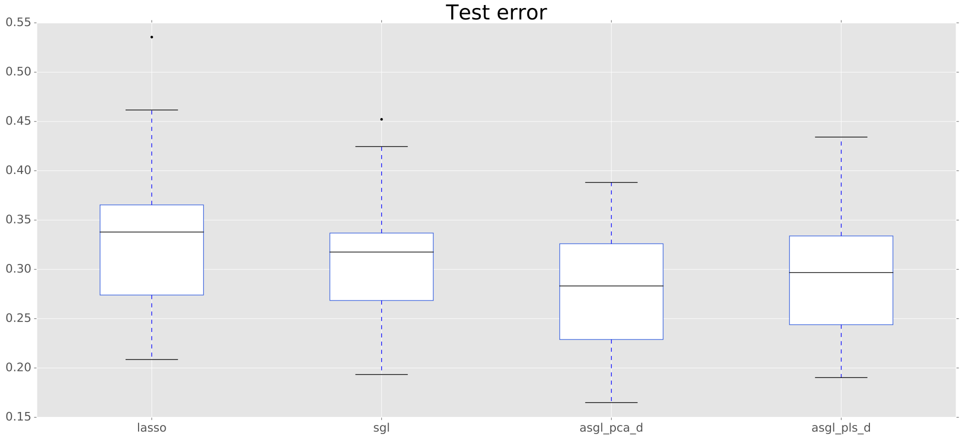
In the case of the dataset used in this section, there are 120 observations from 3734 different genes. The maximum rank of $\boldsymbol{X}$ here is 120, for this reason 120 possible groups are initially considered. Each gene is assigned to the group associated to the principal component in which that gene had its maximum weight. No gene was assigned to one of the groups, and therefore 119 groups averaging 32 genes per group are created this way. It is worth highlighting that the average group size obtained based on this proposal is close to the expected group size in terms of the cytogenetic position. Figure 6 shows a box-plot of the group sizes.

The dataset is randomly divided into 80/20/20 train / validate / test observations and LASSO, SGL, ASGL pls_d and ASGL pca_d models are solved.

Table 3: Gene expression data of rat eye disease. 20 random dataset divisions were considered. Results displayed as mean value, with standard errors in parenthesis.

	E_t	# Variables selected
LASSO	0.34 (0.08)	18.9 (15.4)
SGL	0.31 (0.07)	189.5 (156.6)
ASGL-pca_d	**0.28** (0.06)	56.35 (70.86)
ASGL-pls_d	0.29 (0.06)	101.7 (85.56)

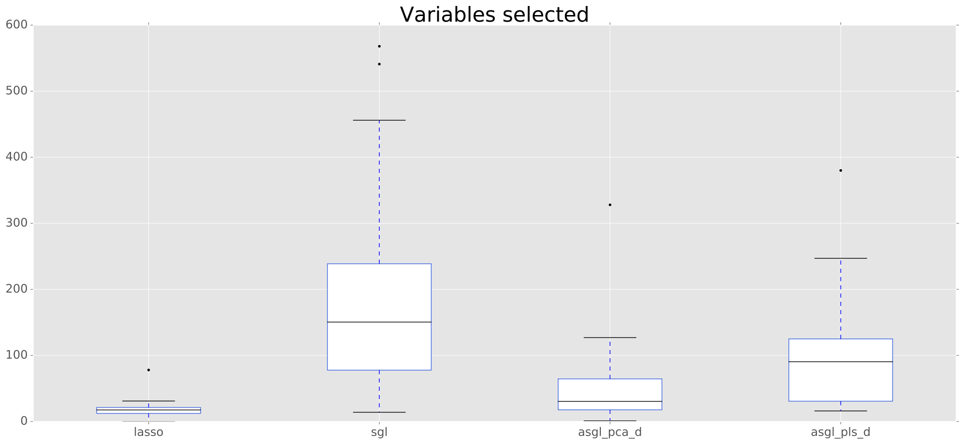
Figure 7: Gene expression data of rat eye disease. 20 random dataset divisions were considered. Box-plot showing the test error.



For each model, the test error E_t and the significant variables selected are obtained. This process is repeated 20 times as a way to gain stability.

The results obtained are shown in Table 3. The best results in terms of the test error are obtained by the proposed ASGL models. LASSO offers a test error approximately 20% greater while SGL test error is 11% greater. Figure 7 displays box-plots of the test error E_t, showing that the spread of E_t is also smaller in the proposed ASGL models providing more stable results. Figure 8 displays box-plots of the number of genes each model selected as significant. The LASSO is the one offering more sparse solutions, using only 19 variables (in mean) per model. SGL is the one using the largest number of variables, approximately 190, and also the one with the largest variability in this metric.

Figure 8: Gene expression data of rat eye disease. 20 random dataset divisions were considered. Box-plot showing the number of significant genes.

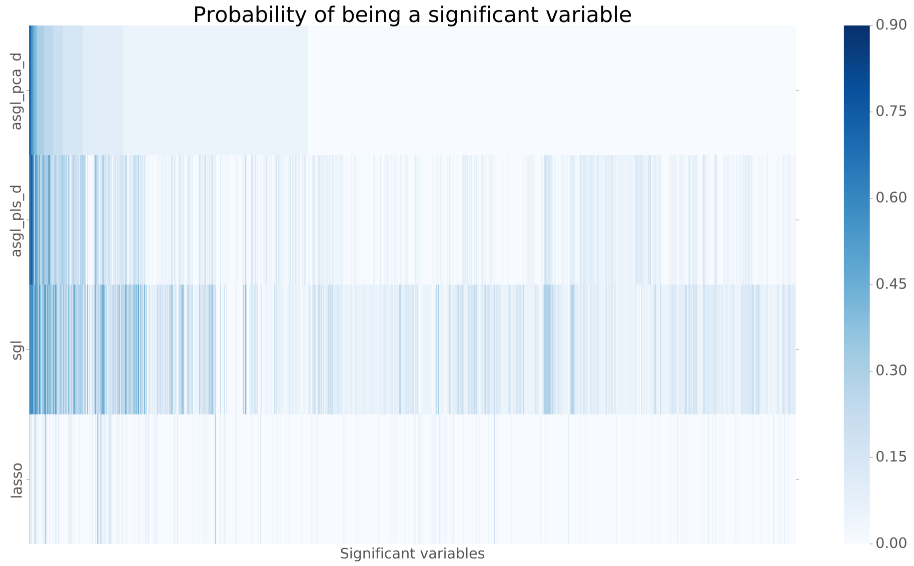


Both ASGL pca_d and ASGL pls_d selected a smaller number of variables than SGL but still larger than LASSO, and they achieve the best prediction results of the four models.

Given that we have the results obtained from 20 repetitions, it is possible to count the number of times each gene has been selected as significant by one of the models in any of the repetitions. Dividing this number by the total number of repetitions, a sort of ”probability of being a significant gene” associated to each gene for each model considered is obtained. Out of the 3734 genes in the dataset, 1612 genes were selected at least one time by any of the models in any of the repetitions (the majority being selected by SGL models). Figure 9 shows the probability of being a significant gene for these 1612 variables and for each model. Rows represent the different models considered and columns represent each gene. Genes are sorted based on the probabilities obtained in the ASGL model with pca_d weights.

Considering a probability threshold of 0.5, only 1 gene in the LASSO models reach a probability of significance above the threshold, showing no stability on the gene selection along the 20 repetitions, and anticipating problems with

Figure 9: Gene expression data of rat eye disease. 20 random dataset divisions were considered. Heatmap showing the probability of being a significant gene. Each row represents a model and each column represents a gene.



possible further biological interpretation of the statistical results. In the case of the SGL model, 35 genes are above the probability threshold, being 0.6 the maximum probability achieved. On the other hand, the ASGL model with pls_d weights includes 17 genes with probabilities above the threshold with a maximum probability value of 0.75, and the ASGL model with pca_d weights has 9 genes above the probability threshold with a maximum probability value of 0.9, showing more stability on the selection along the 20 repetitions and possibly better biological interpretation of the results than the other models.

Results displayed in Table 3 and Figure 9 have been obtained using estimators of the median of the response variable, however, it can be interesting to compare the genes selected at different quantiles. For this reason, the process described above is repeated and LASSO, SGL, ASGL pls_d and ASGL pca_d models are solved for quantile levels $\tau = 0.3$ and $\tau = 0.7$, obtaining probabilities of being a significant gene for each quantile level and each model.

Table 4: Gene expression data of rat eye disease. 20 random dataset divisions were considered. Number of genes above the probability threshold for different quantile levels.

Number of genes above the probability threshold				
	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$	Three quantiles
LASSO	0	1	1	0
SGL	19	35	17	0
ASGL-pca_d	23	9	17	7
ASGL-pls_d	41	17	37	9

Considering a probability threshold of 0.5, Table 4 show the number of genes above the probability threshold for each quantile, and also the number of genes in the same model that have been selected along the different quantile levels.

The LASSO model shows no stability on the variable selection, having only one gene above the threshold for $\tau = 0.5$ and $\tau = 0.7$, and no gene with probability of being significant above 0.5 on the three quantiles simultaneously. The SGL shows some stability across the 20 repetitions considering each quantile independently, but when considering all the quantiles simultaneously it has no gene above the probability threshold. On the other hand, in the case of the ASGL pls_d model, 9 genes had a probability of being significant greater than 0.5 in the 3 quantiles, and in the case of the ASGL pca_d models, 7 genes fulfilled this, showing more robust results than the other estimators.

We conclude that the best results in this real dataset study are provided by the ASGL model with pca_d weights, given that this model is the one with the smallest prediction error and showing great stability on the gene selection.

8 Computational aspects

All the simulations and data analysis commented in Sections 6, and 7 and in the supplementary material were run in a cluster node with two Intel (R)

Xeon(R) CPU E5-2630 v3 (2.4GHz, 20MB Smart Cache) processors, with 32Gb of RAM memory running CentOS 6.5 Final (Rocks 6.1.1 Sand Boa). The computation itself has been developed in Python 2.7.15 (Anaconda Inc.). All the optimization problems have been solved using the CVXPY optimization framework for Python (Diamond and Boyd, 2016) and the open source solver ECOS (Domahidi et al., 2013).

9 Conclusion

In this paper the definition of the SGL estimator has been extended to the QR framework. A new estimator for quantile regression based on the usage of adaptive weights, the adaptive sparse group LASSO in quantile regression has also been proposed. As shown in Section 4, adaptive penalizations are typically centered on the study of the oracle property in both asymptotic and double asymptotic frameworks. A key step on the demonstration of this property is the usage of an initial $\sqrt{n}$-consistent estimator that is usually the result of a nonpenalized model. However, this definition limits the usage of adaptive estimators to low dimensional scenarios. As a solution to this problem, four weight calculation alternatives that can be used in high dimensional scenarios when working with adaptive estimators have been proposed. Section 5.3 conjectures about the relation between these alternatives and the oracle property. Additionally, the performance of the proposed alternatives have been analyzed in a set of synthetic data scenarios that includes high dimensional and low dimensional examples and symmetric error distributions (Section 6). Moreover, a thorough sensitivity analysis studying the behavior of the estimator under different error distributions, and under changes in parameter values has been performed in the supplementary material. The performance of the proposed work is also studied in a real high dimensional dataset including gene expression values of rat eye disease. Previous synthetic data analysis showed that the ASGL estimator is a competitive option in both high and low dimensional scenarios, especially when the adaptive weights are calculated based on subsets of PCA or PLS components. However, when dealing with the real dataset, the ASGL pca_d estimator achieved better results in terms of prediction error and stability of the variables selected. For this reason we conclude that the ASGL pca_d provides the best results among the options proposed in this work.

This work has risen some questions that will require further investigation. One interesting problem is the optimization of the hyper-parameters. In this work we make use of grid-search, but it is worth commenting that new hyper-parameter tuning alternatives have appeared in recent years (Laria et al., 2019), and it can be interesting to investigate the usage of this or other options in the optimization of the parameters of the models introduced in this work.

Section 5.3 has shown some concluding remarks related to the oracle property of the pca_d weight calculation alternative. The pls_d alternative based on PLS, however, is more complex and will require further research. In any case, it is worth mentioning the interesting work performed by Chun and Keleş (2010), that studies the consistency of the PLS estimator in the asymptotic and double asymptotic frameworks, reaching the conclusion (in *Theorem 1*) that given some previous assumptions, if $\frac{p}{n} \to 0$, then

$$\left\|\beta^{PLS} - \beta\right\|_2 \to 0 \text{ in probability.}$$

This result would prove the consistency of the estimator, but It would not be enough for proving the $\sqrt{n}$-consistency, for this reason, we consider that the asymptotic property of the pls_d alternative is a topic for future work.

Finally, simulations from Section 6 have studied different model formulations, including (suggested by a referee) the usage of sparse PCA and sparse PLS in the weight calculation process. The simulations showed that this alternative did not yield to better results than the non sparse PCA or PLS alternatives, but it can be interesting to study other sparse techniques.

10 Acknowledgments

We appreciate the work of the referees that has contributed to substantially improve the scientific contributions of this work. In this research we have made use of Uranus, a supercomputer cluster located at University Carlos III of Madrid and funded jointly by EU-FEDER funds and by the Spanish Government via the National Projects No. UNC313-4E-2361, No. ENE2009-12213- C03-03, No. ENE2012-33219 and No. ENE2015-68265-P. This research was partially supported by research grants and Project ECO2015-66593-P from Ministerio de Economía, Industria y Competitividad, Project MTM2017-88708-P from Ministerio de Economía y Competitividad, FEDER funds and Project IJCI-2017-34038 from Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades.

References

Chatterjee, S., Banerjee, Arindam, S., and Ganguly, A. R. (2011). Sparse Group Lasso for Regression on Land Climate Variables. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 1–8. IEEE.

Chiang, A. P., Beck, J. S., Yen, H.-J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K.-Y. A., Huang, J., Elbedour, K., Carmi, R., Slusarski, D. C., Casavant, T. L., Stone, E. M., and Sheffield, V. C. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet-Biedl syndrome gene (BBS11). *Proceedings of the National Academy of Sciences*, 103(16):6287–6292.

Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 72(1):3–25.

Ciuperca, G. (2017). Adaptive fused LASSO in grouped quantile regression. *Journal of Statistical Theory and Practice*, 11(1):107–125.

Ciuperca, G. (2019). Adaptive group LASSO selection in quantile models. *Statistical Papers*, 60(1):173–197.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *arXiv:1603.00943*.

Domahidi, A., Chu, E., and Boyd, S. (2013). ECOS: An SOCP Solver for Embedded Systems. In *European Control Conference (ECC)*.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Annals of Statistics*, 32(3):928–961.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *ArXiv:1001.0736*, pages 1–8.

Ghosh, S. (2011). On the grouped selection and model complexity of the adaptive elastic net. *Statistics and computing*, 21:451–462.

Huang, J., Horowitz, J. L., and Ma, S. (2008a). Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, 36(2):587–613.

Huang, J., Ma, S., and Zhang, C.-H. (2008b). Adaptive Lasso for Sparse High-dimensional Regression. *Statistica Sinica*, 1(374):1–28.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics: Second Edition*. Wiley Series in Probability and Statistics. wiley, Hoboken, NJ, USA.

Kim, Y., Choi, H., and Oh, H. S. (2008). Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, 103(484):1665–1673.

Koenker, R. (2005). *Quantile Regression*. Cambridge university Press.

Koenker, R. and Bassett, G. (1978). Regression Quantiles. *Econometrica*, 46(1):33–50.

Laria, J. C., Aguilera-Morillo, M. C., and Lillo, R. E. (2019). An iterative sparse-group lasso. *Journal of Computational and Graphical Statistics*, pages 1–21.

Li, Y. and Zhu, J. (2008). L1- -Norm Quantile Regression. *Journal of Computational and Graphical Statistics*, 17(1):1–23.

Loh, P. L. (2017). Statistical consistency and asymptotic normality for high-dimensional robust m-estimators. *Annals of Statistics*, 45(2):866–896.

Nardi, Y. and Rinaldo, A. (2008). On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2(0):605–633.

Poignard, B. (2018). Asymptotic theory of the adaptive Sparse Group Lasso. *Annals of the Institute of Statistical Mathematics*.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C., and Stone, E. M. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

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

Wang, L., Wu, Y., and Li, R. (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association*, 107(497):214–222.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S., and Yan, S. (2010). Sparse Representation for Computer Vision and Pattern Recognition. *Proceedings of the IEEE*, 98(6):1031–1044.

Wu, Y. and Liu, Y. (2009). Variable selection in quantile regression. *Statistica Sinica*, 19(2):801–817.

Yahya Algamal, Z. and Hisyam Lee, M. (2019). A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification. *Advances in Data Analysis and Classification*, 13:753–771.

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

Zhao, W., Zhang, R., and Liu, J. (2014). Sparse group variable selection based on quantile hierarchical Lasso. *Journal of Applied Statistics*, 41(8):1658–1677.

Zhou, N. and Zhu, J. (2010). Group Variable Selection via a Hierarchical Lasso and Its Oracle Property. *Statistics and Its Interface*, 3:557–574.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286.

Adaptive sparse group LASSO in quantile regression

Supplementary material

Álvaro Méndez Civieta[*] [†] M. Carmen Aguilera-Morillo[‡] [†]

Rosa E. Lillo[*] [†]

Simulation study: sensitivity analysis

This supplementary material shows a sensitivity analysis studying the effect of variations on the error distribution of the model as well as different parameters of the ASGL estimator proposed in the article.

0.1 Variation on the model errors

In order to perform well, OLS estimators need to set certain hypothesis on the model errors, namely, being centered, homoscedastic and normally distributed, that are no longer required in quantile regression models. Along this section, the behavior of the proposed ASGL QR estimator is studied under the frame-

[*] Department of Statistics, University Carlos III of Madrid.

[†] uc3m-Santander Big Data Institute.

[‡] Department of Applied Statistics and Operational Research, and Quality, Universitat Politècnica de València

Table 1: Simulation 3. Considering 625 variables and a Cauchy$(0,3)$ error.

	$\left\Vert\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\Vert$	E_t	CSR	TPR	TNR
		625 variables. Sparse distribution of variables			
LASSO	33.69 (4.62)	21.33 (10.53)	**0.87** (0.02)	0.57 (0.08)	**0.91** (0.02)
SGL	25.81 (1.92)	18.43 (10.38)	0.67 (0.12)	**0.89** (0.07)	0.66 (0.13)
ASGL-pca_d	**25.24** (2.08)	**17.89** (10.34)	0.80 (0.05)	0.87 (0.07)	0.79 (0.06)
ASGL-pca_1	25.81 (2.07)	18.40 (10.36)	0.68 (0.14)	**0.89** (0.07)	0.69 (0.15)
ASGL-pls_d	25.47 (2.14)	18.15 (10.33)	0.74 (0.08)	**0.89** (0.06)	0.72 (0.09)
ASGL-pls_1	25.57 (2.16)	18.19 (10.31)	0.75 (0.09)	0.87 (0.006	0.73 (0.10)
		625 variables. Dense distribution of variables			
LASSO	57.52 (16.14)	27.85 (10.71)	**0.95** (0.02)	0.86 (0.06)	**0.96** (0.01)
SGL	26.13 (5.30)	17.65 (8.76)	0.73 (0.13)	**0.99** (0.01)	0.70 (0.15)
ASGL-pca_d	**22.05** (4.90)	**16.25** (8.76)	0.91 (0.11)	**0.99** (0.01)	0.90 (0.13)
ASGL-pca_1	22.65 (5.36)	17.50 (8.78)	0.75 (0.11)	**0.99** (0.01)	0.71 (0.13)
ASGL-pls_d	22.13 (5.07)	16.28 (8.84)	0.89 (0.11)	**0.99** (0.01)	0.88 (0.13)
ASGL-pls_1	22.17 (4.84)	16.28 (8.74)	0.90 (0.09)	**0.99** (0.01)	0.89 (0.10)

work of different error distributions that do not fulfill the OLS hypothesis, showing this way the benefits of the QR formulation.

Simulation 3: Cauchy(0,3) error

In this section the proposed ASGL estimator is studied under the framework of the following model,

$$y = X\beta + \varepsilon,\ \varepsilon \sim \text{Cauchy}(0,3),$$

The main characteristic of the Cauchy distribution is that the central moments in this distribution do not exist, making it an interesting variation on the model error. This distribution is a good example of heavy tail distributions which often appear in practical situations. This simulation show the results obtained under simulation *Case* 1, considering 625 variables sparsely distributed and *Case* 2, considering 625 variables densely distributed.

The results from this simulation scheme are displayed in Table 1. Both

Figure 1: Simulation 3. Sparse distribution of 625 variables. Considering a Cauchy$(0, 3)$ error. Box-plots showing the test error of the different models.

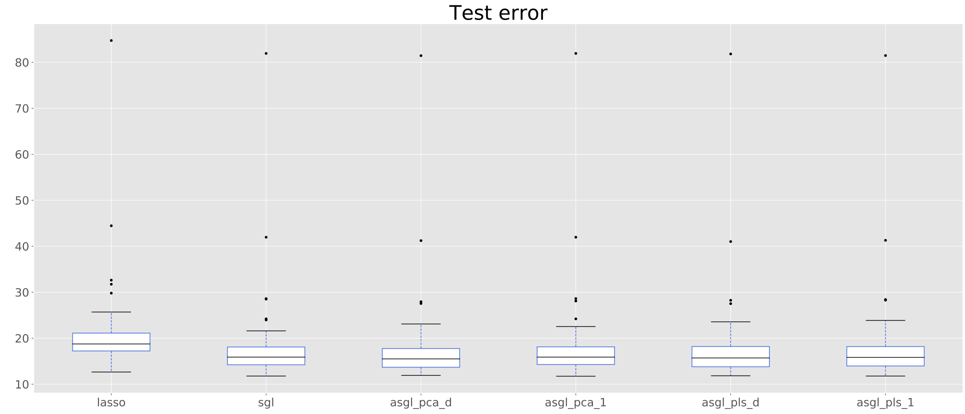
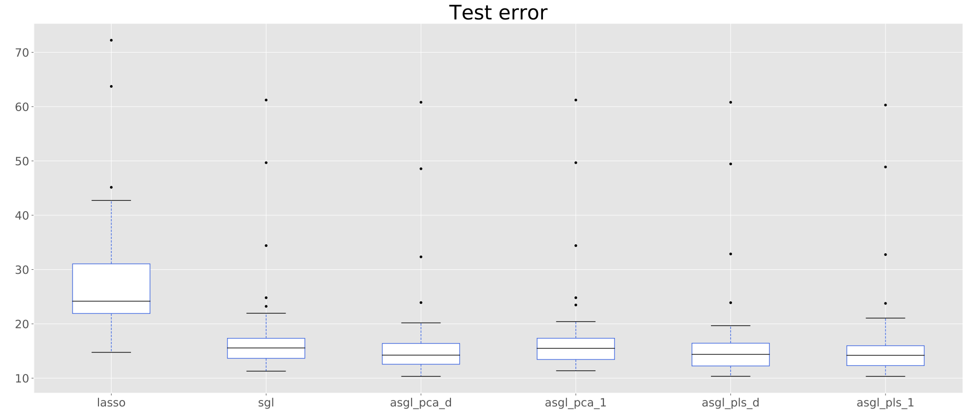


Figure 2: Simulation 3. Dense distribution of 625 variables. Considering a Cauchy$(0, 3)$ error. Box-plots showing the test error of the different models.



in the case of the sparse or the dense distribution of the significant variables, the best results in terms of the distance between predicted and true $\boldsymbol{\beta}$, and the value of E_t are achieved by the proposed ASGL estimator using pca_d weights. The difference in terms of prediction error among models (excepting LASSO, which shows by far the largest error) is smaller in this simulation than in symmetric error ones shown in the article, probably due to the large tails of Cauchy distributions and the associated outliers. However, even under this framework, it is interesting to see that the proposed models offer a good variable selection performance both in terms of TPR and TNR as opposed to lasso (with large TNR but very low TPR) or SGL (with large TPR but low TNR). Figures 1 and 2 display box-plots of the test error value E_t, showing clearly the presence of outliers.

Simulation 4: $\chi^2(3)$ error

In this section the proposed ASGL estimator is studied under the framework of the following model,

$$y = X\beta + \varepsilon, \ \varepsilon \sim \chi^2(3),$$

The χ^2 distribution is non symmetric as opposed to previous error distributions t and Cauchy that were symmetric. This simulation show the results obtained under simulation *Case* 1, considering 625 variables sparsely distributed and *Case* 2, considering 625 variables densely distributed.

The results from this simulation scheme are displayed in Table 2. The best results in terms of the distance between predicted and true $\boldsymbol{\beta}$, and in terms of the test error E_t are obtained by the ASGL model using pca_d weights in the sparse *Case* 1 and pls_d weights in the dense *Case* 2, though both methods

Table 2: Simulation 4. Considering 625 variables and a $\chi^2(3)$ error.

	$\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
		625 variables. Sparse distribution of variables			
LASSO	23.36 (4.00)	7.88 (1.54)	**0.89** (0.01)	0.75 (0.06)	**0.90** (0.01)
SGL	18.97 (2.99)	6.10 (1.00)	0.78 (0.09)	0.88 (0.04)	0.77 (0.10)
ASGL-pca_d	**14.77** (3.19)	**4.62** (0.97)	0.84 (0.04)	0.90 (0.03)	0.83 (0.04)
ASGL-pca_1	18.84 (2.97)	6.07 (1.00)	0.78 (0.07)	0.88 (0.03)	0.77 (0.08)
ASGL-pls_d	15.09 (3.07)	4.71 (0.90)	0.83 (0.04)	**0.91** (0.03)	0.82 (0.04)
ASGL-pls_1	15.09 (3.16)	4.75 (0.99)	0.82 (0.04)	0.90 (0.03)	0.82 (0.04)
		625 variables. Dense distribution of variables			
LASSO	20.06 (11.52)	6.71 (3.88)	**0.95** (0.01)	0.96 (0.03)	**0.95** (0.01)
SGL	8.89 (2.23)	2.80 (0.69)	0.78 (0.10)	**0.99** (0.01)	0.75 (0.12)
ASGL-pca_d	5.79 (1.00)	1.96 (0.28)	0.90 (0.13)	**0.99** (0.01)	0.88 (0.14)
ASGL-pca_1	8.17 (2.30)	2.73 (0.71)	0.80 (0.09)	**0.99** (0.01)	0.77 (0.11)
ASGL-pls_d	**5.75** (1.04)	**1.95** (0.29)	0.89 (0.12)	**0.99** (0.01)	0.87 (0.13)
ASGL-pls_1	5.92 (1.09)	1.99 (0.29)	0.89 (0.08)	**0.99** (0.01)	0.88 (0.09)

Figure 3: Simulation 4. Sparse distribution of 625 variables. Considering a $\chi(3)$ error. Box-plots showing the test error of the different models.

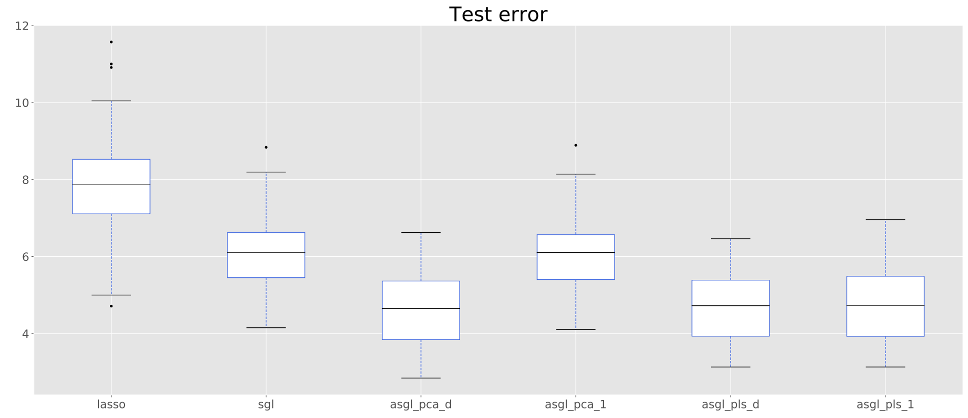
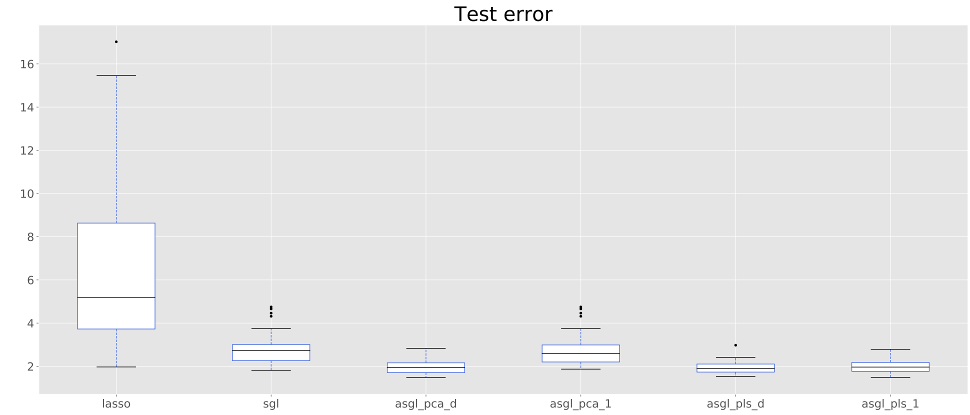


Figure 4: Simulation 4. Dense distribution of 625 variables. Considering a $\chi(3)$ error. Box-plots showing the test error of the different models.



provide quite similar solutions. As in previous simulations, LASSO show a larger TNR value, being the most sparse solution, but also the worst TPR performance, meaning that the selection of significant variables is not very accurate. Opposed to this behavior, SGL show good TPR value but worse TNR, selecting too many non significant variables. The proposed ASGL estimator provides good results both in terms of TPR and TNR. Figures 3 and 4 display box-plots of the test error E_t for the different models, showing that the spread of E_t is much smaller in the ASGL pls_d and pca_d than in the LASSO and SGL (especially in the dense case), indicating that these models provide more stable solutions in terms of prediction accuracy.

0.2 Simulation 5: influence of γ_1 and γ_2

Given equations

$$\tilde{w_j} = \frac{1}{|\hat{\beta}_j|^{\gamma_1}} \quad \text{and} \quad \tilde{v_l} = \frac{1}{\left\|\hat{\boldsymbol{\beta}}^l\right\|_2^{\gamma_2}}, \tag{1}$$

and

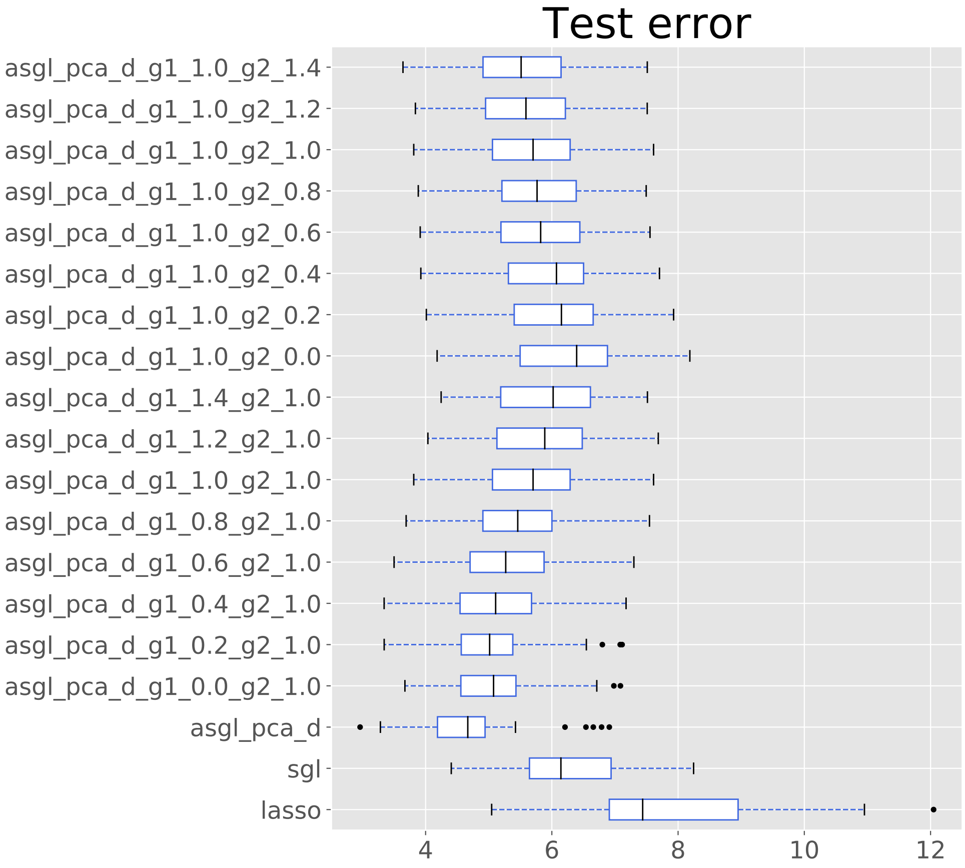
$$\tilde{w_j} = \frac{1}{|q_{1j}|^{\gamma_1}} \quad \text{and} \quad \tilde{v_l} = \frac{1}{\left\|\boldsymbol{q}_1^l\right\|_2^{\gamma_2}}, \tag{2}$$

Table 3: Simulation 5. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of γ_1 and γ_2 influence.

	$\left\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
LASSO	23.88 (4.35)	8.02 (1.60)	**0.88** (0.01)	0.75 (0.06)	**0.90** (0.01)
SGL	19.40 (2.74)	6.19 (0.88)	0.77 (0.07)	0.89 (0.04)	0.76 (0.08)
ASGL	**15.14** (2.97)	**4.66** (0.87)	0.83 (0.03)	**0.92** (0.03)	0.82 (0.04)
$\gamma_1 = 1$ fixed. Varying γ_2					
ASGL-$\gamma_2 = 0.0$	19.74 (2.94)	6.23 (0.94)	0.81 (0.07)	0.89 (0.05)	0.81 (0.08)
ASGL-$\gamma_2 = 0.2$	19.42 (2.97)	6.08 (0.92)	0.72 (0.07)	0.89 (0.05)	0.81 (0.08)
ASGL-$\gamma_2 = 0.4$	19.08 (2.83)	5.95 (0.87)	0.83 (0.05)	0.89 (0.05)	0.82 (0.06)
ASGL-$\gamma_2 = 0.6$	18.74 (2.79)	5.80 (0.85)	0.83 (0.05)	0.89 (0.04)	0.83 (0.05)
ASGL-$\gamma_2 = 0.8$	18.65 (2.97)	5.75 (0.88)	0.84 (0.04)	0.90 (0.04)	0.84 (0.05)
ASGL-$\gamma_2 = 1.0$	18.38 (3.07)	5.66 (0.90)	0.85 (0.04)	0.90 (0.04)	0.85 (0.04)
ASGL-$\gamma_2 = 1.2$	18.24 (3.19)	5.61 (0.94)	0.86 (0.03)	0.90 (0.04)	0.85 (0.04)
ASGL-$\gamma_2 = 1.4$	18.08 (3.32)	5.56 (0.97)	0.87 (0.02)	0.90 (0.04)	0.86 (0.03)
$\gamma_2 = 1$ fixed. Varying γ_1					
ASGL-$\gamma_1 = 0.0$	16.23 (2.79)	5.03 (0.80)	0.80 (0.04)	0.91 (0.03)	0.79 (0.05)
ASGL-$\gamma_1 = 0.2$	16.23 (2.91)	5.03 (0.85)	0.82 (0.04)	0.91 (0.03)	0.81 (0.08)
ASGL-$\gamma_1 = 0.4$	16.54 (2.93)	5.12 (0.87)	0.84 (0.03)	0.91 (0.03)	0.83 (0.04)
ASGL-$\gamma_1 = 0.6$	17.07 (2.94)	5.28 (0.88)	0.84 (0.04)	0.90 (0.04)	0.84 (0.04)
ASGL-$\gamma_1 = 0.8$	17.69 (2.96)	5.46 (0.89)	0.85 (0.03)	0.90 (0.04)	0.84 (0.04)
ASGL-$\gamma_1 = 1.0$	18.38 (3.07)	5.66 (0.90)	0.85 (0.03)	0.90 (0.04)	0.85 (0.04)
ASGL-$\gamma_1 = 1.2$	18.90 (3.07)	5.81 (0.89)	0.85 (0.04)	0.90 (0.05)	0.84 (0.05)
ASGL-$\gamma_1 = 1.4$	19.47 (3.02)	5.96 (0.86)	0.85 (0.04)	0.90 (0.05)	0.85 (0.04)

for the calculation of the weights, one can see that the formulation includes two nonnegative parameters, γ_1 in the lasso weights part and γ_2 in the group lasso weights part that are the powers entering the weights. Along this section a simulation studying the influence of the value of these parameters is performed. The simulation scheme is that of *Case 1*: 625 variables sparsely distributed. Additionally, a t(3) distribution error is considered, and the weights are calculated based on a subset of PCA components pca_d. Two situations are studied: the behavior of the ASGL estimator while varying the value of γ_2 and leaving $\gamma_1 = 1$ and the behavior of the ASGL estimator while varying the value of γ_1 and leaving $\gamma_2 = 1$. The results are compared against the LASSO, SGL and the ASGL estimator optimizing both γ_1 and γ_2.

Figure 5: Simulation 5. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of γ_1 and γ_2 influence. Box-plots showing the test error of the different models.



The results obtained in this simulation are displayed in Table 3 and Figure 5. The best results in terms of the distance between predicted and true $\boldsymbol{\beta}$, and the value of E_t are provided by the ASGL estimator while optimizing both γ_1 and γ_2, highlighting the importance of the selection of this parameters. It is also interesting to observe how, while fixing γ_1, errors decrease as γ_2 increase, but while fixing γ_2, the opposite behavior appears, and errors increase as γ_1 increase.

0.3 Influence of $\alpha_{pca,d}$ and $\alpha_{pls,d}$

The weight calculation alternatives pca_d and pls_d are based on selecting a subset of d either PCA or PLS components that explain up to a certain percentage of variability, $\alpha_{pca,d}$ or $\alpha_{pls,d}$ respectively, priorly fixed by the researcher. Along this section, the effect of changes in the percentage of explained variability is studied. The simulation schemes are these of *Case 1* (625 variables sparsely distributed) and *Case 5* (100 variables sparsely distributed). Additionally, a t(3) distribution error is considered. Finally, two situations will be studied: variations on the percentage of variability affecting pca_d technique and variations on the percentage of variability affecting pls_d technique.

Simulation 6: Influence of $\alpha_{pca,d}$

This simulation is centered on the effect of variations in the percentage of explained variability using PCA. Since PCA technique defines an orthogonal change of basis matrix, it is possible to recover all the variability from the original variables, and thus, different ASGL pca_d models are solved ranging the percentage of explained variability from 10% to 100%.

The results obtained are shown in Table 4. In the low dimensional frame-

Table 4: Simulation 6. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pca,d}$ influence.

	$\left\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
	625 variables. Sparse distribution of variables.				
LASSO	21.85 (4.77)	7.40 (1.77)	0.89 (0.07)	0.77 (0.07)	0.90 (0.08)
SGL	18.14 (3.28)	5.80 (1.07)	0.80 (0.06)	0.89 (0.05)	0.79 (0.10)
ASGL-*pca* − 10%	17.96 (3.32)	5.76 (1.09)	0.80 (0.08)	0.89 (0.05)	0.79 (0.09)
ASGL-*pca* − 20%	17.54 (3.47)	5.60 (1.13)	0.81 (0.07)	0.89 (0.05)	0.80 (0.07)
ASGL-*pca* − 30%	17.54 (3.45)	5.60 (1.12)	0.82 (0.06)	0.90 (0.05)	0.79 (0.09)
ASGL-*pca* − 40%	16.73 (3.78)	5.33 (1.22)	0.84 (0.04)	0.90 (0.04)	0.80 (0.08)
ASGL-*pca* − 50%	15.47 (3.78)	4.92 (1.25)	0.84 (0.04)	0.90 (0.04)	0.82 (0.07)
ASGL-*pca* − 60%	13.35 (3.47)	4.15 (1.16)	0.84 (0.04)	0.92 (0.04)	0.83 (0.05)
ASGL-*pca* − 70%	12.76 (3.37)	3.92 (1.04)	0.84 (0.04)	0.93 (0.04)	0.83 (0.05)
ASGL-*pca* − 80%	12.98 (3.36)	4.01 (1.02)	0.84 (0.04)	0.93 (0.04)	0.83 (0.04)
ASGL-*pca* − 90%	13.04 (3.41)	4.04 (1.03)	0.84 (0.04)	0.92 (0.04)	0.84 (0.04)
ASGL-*pca* − 100%	14.08 (3.76)	4.34 (0.16)	0.84 (0.03)	0.92 (0.04)	0.84 (0.03)
	100 variables. Sparse distribution of variables.				
LASSO	0.58 (0.08)	0.59 (0.01)	0.73 (0.01)	1.00 (0.00)	0.66 (0.14)
SGL	0.60 (0.08)	0.59 (0.01)	0.72 (0.12)	1.00 (0.00)	0.57 (0.17)
ASGL-*pca* − 10%	0.59 (0.07)	0.59 (0.01)	0.83 (0.10)	1.00 (0.00)	0.60 (0.14)
ASGL-*pca* − 20%	0.60 (0.07)	0.59 (0.01)	0.75 (0.10)	1.00 (0.00)	0.60 (0.17)
ASGL-*pca* − 30%	0.59 (0.07)	0.59 (0.01)	0.78 (0.10)	1.00 (0.00)	0.61 (0.14)
ASGL-*pca* − 40%	0.58 (0.07)	0.59 (0.01)	0.79 (0.10)	1.00 (0.00)	0.64 (0.14)
ASGL-*pca* − 50%	0.56 (0.07)	0.58 (0.01)	0.78 (0.10)	1.00 (0.00)	0.68 (0.13)
ASGL-*pca* − 60%	0.55 (0.08)	0.58 (0.01)	0.79 (0.10)	1.00 (0.00)	0.70 (0.14)
ASGL-*pca* − 70%	0.55 (0.07)	0.58 (0.01)	0.78 (0.11)	1.00 (0.00)	0.69 (0.17)
ASGL-*pca* − 80%	0.54 (0.07)	0.58 (0.01)	0.79 (0.10)	1.00 (0.00)	0.70 (0.16)
ASGL-*pca* − 90%	0.52 (0.07)	0.58 (0.01)	0.82 (0.11)	1.00 (0.00)	0.74 (0.17)
ASGL-*pca* − 100%	0.44 (0.05)	0.57 (0.01)	0.94 (0.07)	1.00 (0.00)	0.92 (0.10)

Figure 6: Simulation 6. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pca,d}$ influence. Box-plots showing the test error of the different models.

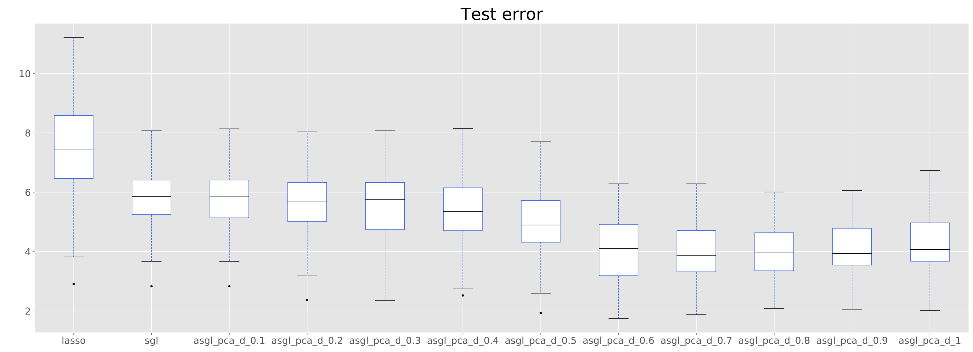


Figure 7: Simulation 6. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pca,d}$ influence. Box-plots showing the correct selection rate of the different models.

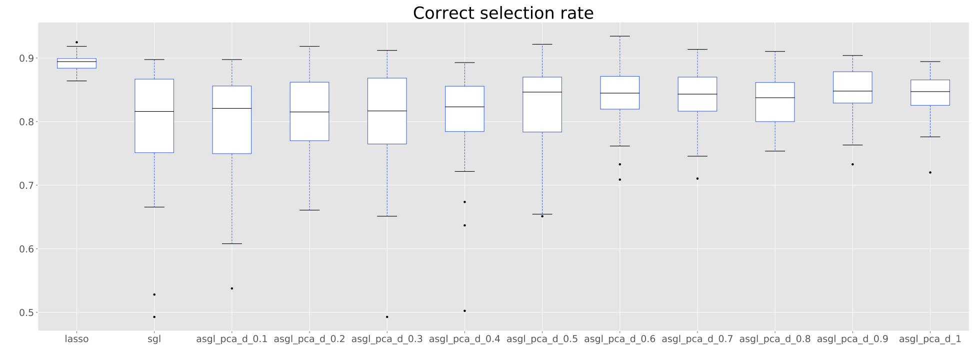


Figure 8: Simulation 6. Sparse distribution of 100 variables. Considering a t(3) error. Analysis of $\alpha_{pca,d}$ influence. Box-plots showing the test error of the different models.

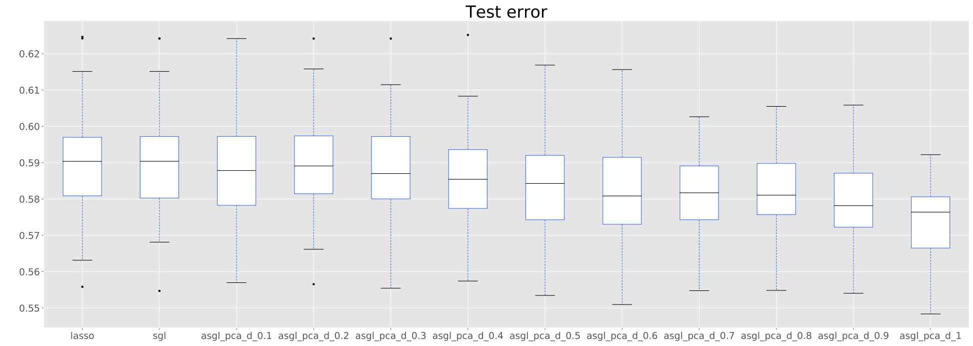
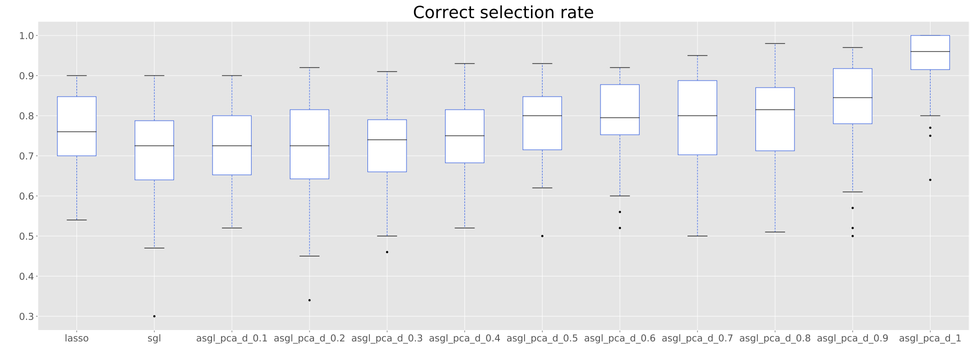


Figure 9: Simulation 6. Sparse distribution of 100 variables. Considering a t(3) error. Analysis of $\alpha_{pca,d}$ influence. Box-plots showing the correct selection rate of the different models.



work considering 100 variables it is possible to see how as the percentage of variability increases, all the metrics are improved achieving smaller prediction errors and better variable selection. A similar behavior is observed in the high dimensional framework for the explained variability ranging between 10% up to, approximately, 80%. However, when further increasing the percentage of explained variability up to 100%, the results get worse. Our guess is that in high dimensional frameworks, attaining a 100% of explained variability in PCA requires obtaining as many principal components as rows in the data matrix, producing overfitted solutions and adding noise to the predictions. Figures 6 and 7 show boxplots of the prediction error E_t and the correct selection rate in the high dimensional framework, while Figures 8 and 9 show the same boxplots in the low dimensional framework. In these boxplots the behavior described above can be easily seen.

Simulation 7: Influence of $\alpha_{pls,d}$

This simulation is focused on the effect of variations in the percentage of explained variability using PLS. PLS defines a non-necesarily orthogonal change of basis matrix, and therefore, it is not possible to recover all the variability from the original variables. Actually, in the scheme considering 100 variables, PLS technique could recover at most 70% of the original variabiity, while in the simulation scheme considering 625 variables, PLS could recover at most 60%. For this reason, in the low dimensional framework different ASGL pls_d models are solved ranging the percentage of explained variability from 10% to 70%, while in the high dimensional framework the variability ranges from 10% to 60%.

The results obtained in this simulation are shown in Table 5. In the low di-

Table 5: Simulation 7. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pls,d}$ influence.

	$\left\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\right\|$	E_t	CSR	TPR	TNR
	625 variables. Sparse distribution of variables.				
LASSO	23.66 (4.97)	7.99 (1.82)	0.85 (0.04)	0.76 (0.07)	0.90 (0.01)
SGL	18.63 (3.95)	6.06 (1.35)	0.84 (0.04)	0.90 (0.04)	0.79 (0.08)
ASGL-$pls - 10\%$	13.88 (4.23)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.84 (0.04)
ASGL-$pls - 20\%$	14.19 (4.20)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.83 (0.04)
ASGL-$pls - 30\%$	14.19 (4.20)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.83 (0.04)
ASGL-$pls - 40\%$	14.19 (4.20)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.83 (0.04)
ASGL-$pls - 50\%$	14.19 (4.20)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.83 (0.04)
ASGL-$pls - 60\%$	14.19 (4.20)	4.42 (1.30)	0.84 (0.04)	0.92 (0.03)	0.83 (0.04)
	100 variables. Sparse distribution of variables.				
LASSO	0.60 (0.07)	0.60 (0.01)	0.77 (0.09)	1.00 (0.00)	0.67 (0.13)
SGL	0.60 (0.07)	0.60 (0.01)	0.73 (0.12)	1.00 (0.00)	0.90 (0.12)
ASGL-$pls - 10\%$	0.50 (0.07)	0.58 (0.01)	0.87 (0.09)	1.00 (0.00)	0.92 (0.13)
ASGL-$pls - 20\%$	0.46 (0.06)	0.58 (0.01)	0.93 (0.08)	1.00 (0.00)	0.93 (0.12)
ASGL-$pls - 30\%$	0.45 (0.06)	0.57 (0.01)	0.94 (0.08)	1.00 (0.00)	0.93 (0.11)
ASGL-$pls - 40\%$	0.45 (0.06)	0.57 (0.01)	0.95 (0.07)	1.00 (0.00)	0.93 (0.11)
ASGL-$pls - 50\%$	0.45 (0.06)	0.57 (0.01)	0.95 (0.07)	1.00 (0.00)	0.93 (0.09)
ASGL-$pls - 60\%$	0.45 (0.06)	0.57 (0.01)	0.96 (0.05)	1.00 (0.00)	0.95 (0.07)
ASGL-$pls - 70\%$	0.45 (0.06)	0.57 (0.01)	0.96 (0.05)	1.00 (0.00)	0.95 (0.07)

Figure 10: Simulation 7. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pls,d}$ influence. Box-plots showing the test error of the different models.

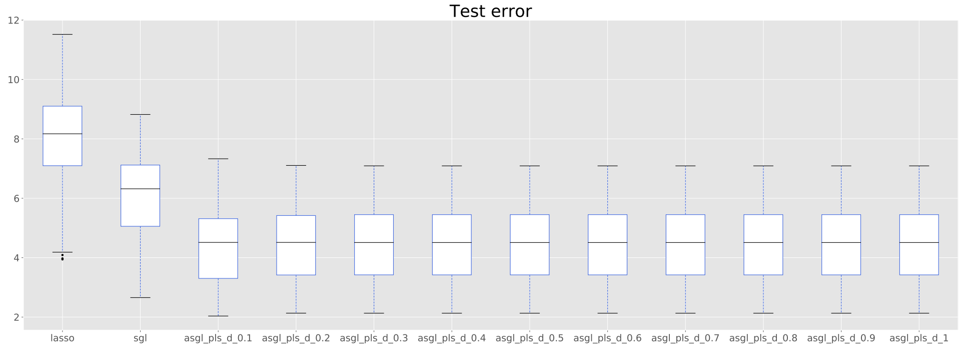


Figure 11: Simulation 7. Sparse distribution of 625 variables. Considering a t(3) error. Analysis of $\alpha_{pls,d}$ influence. Box-plots showing the correct selection rate of the different models.

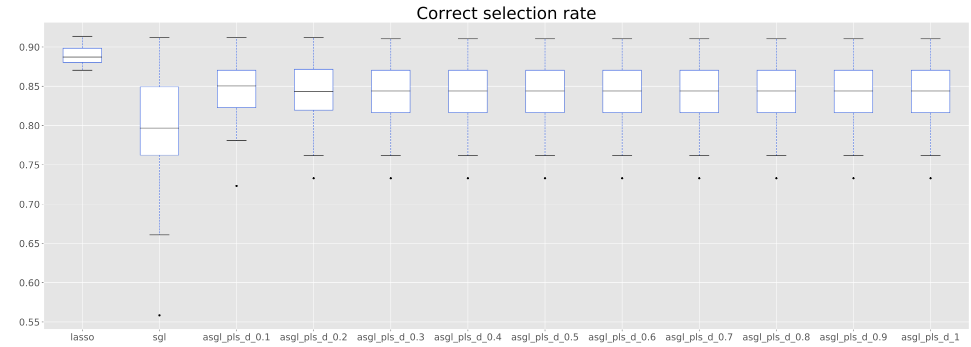


Figure 12: Simulation 7. Sparse distribution of 100 variables. Considering a t(3) error. Analysis of $\alpha_{pls,d}$ influence. Box-plots showing the test error of the different models.

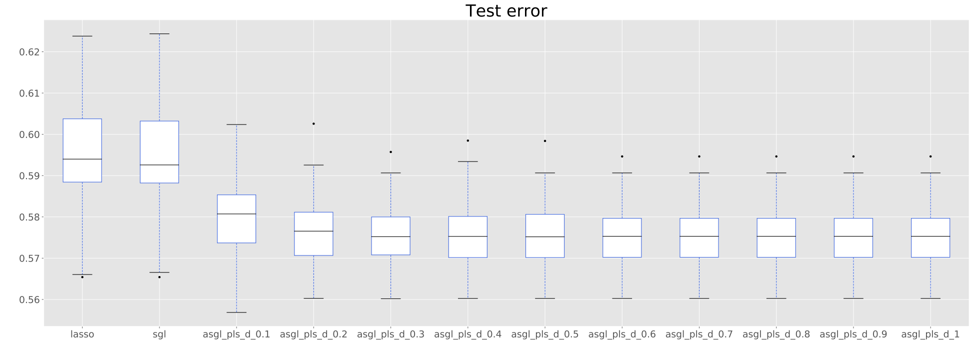
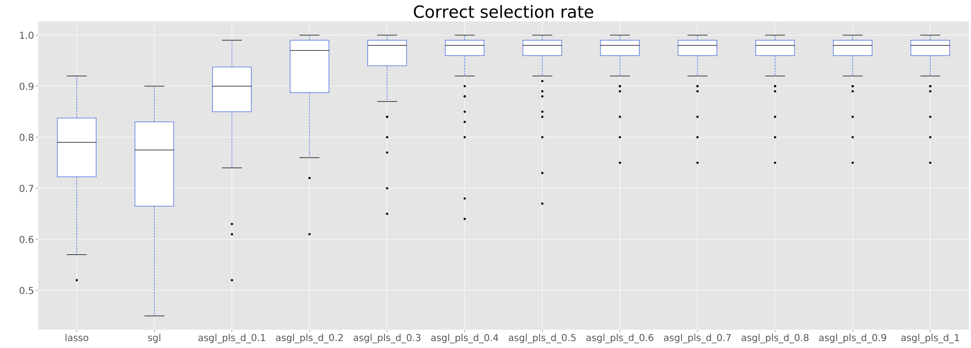


Figure 13: Simulation 7. Sparse distribution of 100 variables. Considering a t(3) error. Analysis of $\alpha_{pls,d}$ influence. Box-plots showing the correct selection rate of the different models.



mensional framework considering 100 variables it is possible to see how as the percentage of variability increases from 10% up to 30% results improve slightly in terms of prediction accuracy. Further increases up to 70% produce small improvements in the TNR, but overall, changes in the percentage of explained variability in PLS do not affect heavily the performance of the estimator. This is probably due to the way the PLS components are obtained, based on the maximization of the covariance between the response variable and the covariates. This means that the first PLS components already hold the information most related to the response variable, providing very good results. A similar behaviour is observed in the high dimensional fraework, where the prediction accuracy stabilizes while considering a 20% of explained variability. Figures 10 and 11 show boxplots of the prediction error E_t and the correct selection rate in the high dimensional framework, while Figures 12 and 13 show the same boxplots in the low dimensional framework. In these boxplots the behavior described above can be easily seen.