An Algorithm for Iterative Selection of Blocks of Features

Pierre Alquier

[1] LPMA (University Paris 7)
175, rue du Chevaleret
75013 Paris FRANCE
alquier@math.jussieu.fr
http://alquier.ensae.net/
[2] CREST (ENSAE)

Abstract. We focus on the problem of linear regression estimation in high dimension, when the parameter β is "sparse" (most of its coordinates are 0) and "blocky" (β_i and β_{i+1} are likely to be equal). Recently, some authors defined estimators taking into account this information, such as the Fused-LASSO [19] or the S-LASSO [10] among others. However, there are no theoretical results about the obtained estimators in the general design matrix case. Here, we propose an alternative point of view, based on the Iterative Feature Selection method [1]. We propose an iterative algorithm that takes into account the fact that β is sparse *and* blocky, with no prior knowledge on the position of the blocks. Moreover, we give a theoretical result that ensures that every step of our algorithm actually improves the statistical performance of the obtained estimator. We provide some simulations, where our method outperforms LASSO-type methods in the cases where the parameter is sparse and blocky. Moreover, we give an application to real data (CGH arrays), that shows that our estimator can be used on large datasets.

Keywords: Feature Selection, Sparsity, Linear Regression, Grouped Variables, ArrayCGH.

1 Introduction

1.1 Setting of the Problem

We assume that we are in the gaussian linear regression setting

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \sim \mathcal{N}\left(X\beta, \begin{pmatrix} \sigma^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2 \end{pmatrix} \right) \tag{1}$$

where X is some (deterministic) real-valued matrix $n \times p$ and $\beta = (\beta_1, ..., \beta_p)' \in \mathbb{R}^p$, with possibly $p > n$ (we use the convention that any $u \in \mathbb{R}^p$ is a column vector and we use the notation u' for its transpose). Let X_1, ..., X_p be the columns

of X, and let $\mathbf{P}$ denote the probability distribution of Y given by Equation 1 (for the sake of simplicity, we assume that the data are normalized in such a way that $X_j' X_j = n$). Our purpose is to estimate β, based on the observation of both X and Y. We use as a criterion the quadratic loss, for any $b \in \mathbb{R}^p$, we put

$$L(b) = \|X(b-\beta)\|_2^2 .$$

It is well known that, as soon as $p > n$, it is not possible to build an estimator with a small loss $L(\cdot)$ - unless some additional condition is satisfied. For example, one may assume that β is sparse, which means that the number of non-zero coefficients in β (usually refered as $\|\beta\|_0$) is small. If $\|\beta\|_0 \ll n$, the LASSO estimator [18], the Dantzig selector [6] or the nonnegative garrote [4], among others, may achieve good performance: see for example the simulations in [18], and the theoretical results in [5] and [3]. Also note that [21] provides a nice survey of the theoretical results for the LASSO and the conditions needed to prove these theoretical results.

Here, we focus on the case where β is sparse *and blocky*. By this, we mean that the non-zero coefficients of β are "grouped" in blocks where they have the same value. For example, a typical parameter β could be

$$\beta = (0,0,0,5,5,5,0,0,0,0,0,1,1,1,2,2,0,0)' .$$

An example of application is given in genomics, where models like the one in (1) may appear. The observation Y_i represents a characteristic of a tumor and depends on some lesions appearing on the chromosomes of cancer cells $X_{j,i}$, the index j representing the localization on the chromosome. Since a lesion generally affects a whole part of a chromosome (duplication or deletion of a region), one may expect that two consecutive parameters β_j and β_{j+1} have some strong relationship. For example, [15] use such an assumption - the main difference is that [15] considers the context of classification, and not of regression. They use a Support Vector Machine with a penalization of the type

$$\lambda \sum_{j=1}^{p} |\beta_j| + \mu \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|,$$

called Fused-SVM in the paper. Using such a penalisation, we expect to find a sparse and blocky solution: the first term ensures sparsity when λ is large, the second term ensures that for most j, $\beta_j = \beta_{j-1}$ and so there are blocks of similar values in the solution. Such a penalization is also used in [12] for genomic applications, with a logistic regression model.

Some estimators have been proposed for the regression case. Let us mention the Fused-LASSO [19], say $\tilde{\beta}^F_{s,t}$, given by

$$\min_b \left\{ \|Y - Xb\|_2^2 + 2ns \sum_{j=1}^{p} |\beta_j| + 2nt \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \right\}$$

with $s, t > 0$, or the S-LASSO estimator, say $\tilde{\beta}^S_{s,t}$, by

$$\min_b \left\{ \|Y - Xb\|_2^2 + 2ns \sum_{j=1}^{p} |\beta_j| + 2nt \sum_{j=2}^{p} (\beta_j - \beta_{j-1})^2 \right\}$$

proposed in [9] and studied in [10] (the S-LASSO does not actually lead to blocky solutions, but to smooth solutions: most $\beta_j - \beta_{j-1}$ are small, but not necessarily 0). See also the structured feature selection in [17]. These estimators can be approximated in practice even for large p: for example the Pathwise Coordinate Optimization algorithm [8] can be used for the S-LASSO, and is used for the Fused-LASSO in [8] when $X = I_n$ (from now I_k will denote the indentity matrix of size k). The Fused-LASSO can be approximated in the general case using for example the algorithms in [11] and [20]. However, note that no general theoretical results were provided in order to estimate a sparse blockwise β. Hebiri [9] provides good guarantees for the S-LASSO under the sparsity assumption but does not take any advantage of the smooth aspect of β, if any. Rinaldo [16] gives theoretical guarantees for the Fused-LASSO that can be applied only in the case where $X = I_n$, and so $n = p$.

Finally, the Group-LASSO introduced by [22] has very interesting practical performance, as well as good theoretical properties studied in [7]. See also [23]. However, this procedure requires the prior knowledge of the location of the groups. The S-LASSO and Fused-LASSO do not require such a prior, and in this paper, we are interested in the case where we do not have this knowledge. Our setting is then:

- we know that most of the $\beta_j = 0$ but we do not know the j's such that $\beta_j \neq 0$;
- we know that most of the $\beta_j = \beta_{j+1}$ but we do not know the j's such that $\beta_j \neq \beta_{j+1}$.

1.2 Overview of the Paper

In this paper, we propose an algorithm to estimate a sparse and blockwise β in the model given by (1), without prior knowledge on the location of the groups. This algorithm is an iterative algorithm. It starts from the initial value $\hat{\beta}^{(0)} = (0, \ldots, 0)$. Then, at each step m, we compute $\hat{\beta}^{(m+1)}$ from $\hat{\beta}^{(m)}$ in the following way: a particular coordinate or group of consecutive coordinates is selected, and then updated. The way to update this coordinate, or group of coordinates, is described in Section 2. The way to select the coordinate, or the group, is postponed to Section 3.

Actually, in Section 3, we give the following result: with large probability, at each step, whatever the choice of the coordinate (or group) to be updated, $L(\hat{\beta}^{(m+1)}) \leq L(\hat{\beta}^{(m)})$. This is Theorem 1. A refinement of this result, Theorem 2, allows to choose the particular coordinate (or group) that ensures the largest possible decrease from $L(\hat{\beta}^{(m)})$ to $L(\hat{\beta}^{(m+1)})$.

We then provide a simulation study in Section 4 with a comparison to the S-LASSO and the Fused-LASSO estimators. Our method outperforms the LASSO-type estimators when the parameter is sparse and blocky. We also provide an application to real data (arrayCGH) in Section 5.

Finally, the proof of Theorems 1 and 2 are given in Section 6.

2 Construction of our Estimator

We now describe our algorithm. It is an iterative algorithm, that starts from $\hat{\beta}^{(0)} = (0, \ldots, 0)$. At each step, we are going to compute $\hat{\beta}^{(m+1)}$ from $\hat{\beta}^{(m)}$. Let us define the soft thresholding function

$$\forall x \in \mathbb{R}, \forall u \geq 0, \quad \gamma(x,u) = \text{sign}(x)(|x| - u)_+$$

where $\text{sign}(x)$ is the sign of x. For any given $j \in \{1, ..., p\}$ and $k \in \{1, ..., p-j+1\}$ let us define $\mathbf{1}_{j,k} \in \mathbb{R}^p$ by $(\mathbf{1}_{j,k})_i = 1$ if $i \in \{j, ..., j+k-1\}$, and $(\mathbf{1}_{j,k})_i = 0$ otherwise (in other words, $\mathbf{1}_{j,k} \in \mathbb{R}^p$ is the pattern of the group of variables $(j, j+1, \ldots, j+k-1)$, of length k).

At each step, several possible moves are considered in our algorithm: a possible move is to update a coordinate or a group of coordinates of maximal size K, for a given $K \in \{1, ..., p\}$. Let us now describe the move for a chosen coordinate or group of coordinates. The way to choose what group of coordinates is to be updated is discussed in the next section, in view of some theoretical results. Let us choose $s > 0$ (the value of s is also discussed in the next section).

Parameter update Let us assume that we are going to update the group of coordinates $(j, j+1, ..., j+k-1)$ (a single coordinate is a particular case with $k = 1$). We define $\hat{\beta}^{(m+1)}$ by

$$\left\{ \begin{array}{c} \hat{\beta}_j^{(m+1)} = \hat{\beta}_j^{(m)} + \gamma\left(\frac{(Y - X\hat{\beta}^{(m)})' \sum_{h=j}^{j+k-1} X_h}{\mathbf{1}_{j,k}' X'X \mathbf{1}_{j,k}}, \frac{s}{\sqrt{\mathbf{1}_{j,k}' X'X \mathbf{1}_{j,k}}} \right) \\ \vdots \\ \hat{\beta}_{j+k-1}^{(m+1)} = \hat{\beta}_{j+k-1}^{(m)} + \gamma\left(\frac{(Y - X\hat{\beta}^{(m)})' \sum_{h=j}^{j+k-1} X_h}{\mathbf{1}_{j,k}' X'X \mathbf{1}_{j,k}}, \frac{s}{\sqrt{\mathbf{1}_{j,k}' X'X \mathbf{1}_{j,k}}} \right) \end{array} \right. \tag{2}$$

and $\hat{\beta}_\ell^{(m+1)} = \hat{\beta}_\ell^{(m)}$ for any $\ell \notin \{j, ..., j+k-1\}$.

Observe that in the case where $k = 1$, this expression is very natural:

$$\hat{\beta}_j^{(m+1)} = \hat{\beta}_j^{(m)} + \gamma\left(\frac{(Y - X\hat{\beta}^{(m)})' X_j}{n}, s \right)$$

and $\hat{\beta}_\ell^{(m+1)} = \hat{\beta}_\ell^{(m)}$ for any $\ell \neq j$. If we take $K = 1$, so we never consider groups of coordinates, we obtain the Iterative Feature Selection algorithm studied in [1].

3 Main Result and Comments

We now give some properties satisfied by any algorithm using only the kind of moves described in the previous section. These results will also provide us a rule to choose the group of coordinates that have to be updated at step m, to calibrate the parameters s and K and, finally, to decide at which step m to stop our iterations.

Note that the proof of the following result is given at the end of the paper, in Section 6, page 10.

Theorem 1 *Let us assume that, as described previously, we start from* $\hat{\beta}^{(0)} = (0, \dots, 0)$*, that at each step* m *we update one group of coordinates following Equation 2 (the choice of the group* **may** *be data-driven). We assume that we stop after* M *steps. We have*

$$\mathbf{P}\left[L(\hat{\beta}^{(M)}) \leq L(\hat{\beta}^{(M-1)}) \leq \cdots \leq L(\hat{\beta}^{(0)})\right] \geq 1 - Kpe^{-\frac{s^2}{2\sigma^2}}.$$

In a way, this result states that every strategy using only the kind of moves described in Section 2 cannot result in overfitting: we are "almost certain" to get closer to our (unknown) objective β at each step. We can quantify this "almost certain": it means "with probability at least $1 - \varepsilon$", if we choose s such that $Kp\exp(-ns^2/(2\sigma^2)) \leq \varepsilon$. For example,

$$s = \sqrt{2\sigma^2 \log\left(\frac{pK}{\varepsilon}\right)}$$

gives a confidence level $1 - \varepsilon$. Note that it is very similar to the theoretical value for the regularization parameter in the LASSO proposed in [3]. This link will be made clearer in the conclusion of the paper.

Note that if we allow $K = p$ (to test every group of size 1 to p), this will not make a big difference because the threshold s will become

$$s = \sqrt{2\sigma^2 \log\left(\frac{p^2}{\varepsilon}\right)} \leq \sqrt{4\sigma^2 \log\left(\frac{p}{\varepsilon}\right)}.$$

However, when p is large, it could be more convenient to choose a smaller K for algorithmic reasons. Let us stress this conclusion for K: $K = p$ is probably almost optimal in theory, but if p is large, we may want to choose a smaller K to keep the computation time small (we will shortly discuss this point in Subsection 3.1).

Also, note that this choice for s is not necessarily the best choice in practice: it requires the knowledge of σ^2, and it is usually too large (see the experiments in [2] for the case $K = 1$). Moreover it is not data-driven. Cross-validation will provide better results in practice.

Finally, we still do not know "how much closer to β" we can move at every step, and we still have no idea of how to choose if we are going to update 1, 2 ... or K coordinates, and wich coordinates we are going to update - we just know that in some sense, every choice is allowed. We now make some propositions.

Sequential strategy Just sequentially update $j = 1$, $j = 2$, ..., $j = p$ with 1 coordinates moves, then $(j, j+1) = (1, 2)$, ..., $(j, j+1) = (p-1, p)$ with 2 coordinates moves, then $(j, j+1, j+2) = (1, 2, 3)$ ... with 3 coordinates moves, ... up to K coordinates moves, and start again. This strategy (with $K = 2$) is in some way similar to the one proposed for example in [8] for an algorithm computing the Fused-LASSO estimate.

Now, a finer version of Theorem 1 will give us an opportunity to make a data-driven choice of the group of coordinates to update at each step.

Theorem 2 *In the same setting as for Theorem 1,*

$$\mathbf{P}\left[\forall m \in \{1, ..., M\} : L(\hat{\beta}^{(m)}) \leq L(\hat{\beta}^{(m-1)}) - \|X(\hat{\beta}^{(m)} - \hat{\beta}^{(m-1)})\|_2^2\right] \geq 1 - Kpe^{-\frac{s^2}{2\sigma^2}}.$$

Note that the proof of Theorem 2 will be included in the proof of Theorem 1.

"Best single move" strategy We propose, at each step, to choose the move that maximises $\|X(\hat{\beta}^{(m)} - \hat{\beta}^{(m-1)})\|_2^2$. Moreover, this allows to define a stopping criterion: we stop when all the possible moves give an improvement $\|X(\hat{\beta}^{(m)} - \hat{\beta}^{(m-1)})\|_2^2 < 1/n^2$ for example.

Note that in any case we do not claim that this strategy is the best possible one. It is possible that iterations gets stuck in some regions of $\mathbb{R}^p$ that are not the most interesting ones (in the same way that sequential optimization algorithms may be trapped in some local minimum). A way to avoid this risk is not clear yet. This is actually the object of an alternative estimator proposed in the conclusion of this paper: see Equation 5 page 14. This estimator will be the object of a future work. However, there are two things that we know for sure about this strategy:

1. Theorem 2 ensures that, after m steps, with probability at least $1 - Kpe^{-\frac{s^2}{2\sigma^2}}$,
$$L(\hat{\beta}^{(m)}) \leq L(\hat{\beta}^{(0)}) - \sum_{k=1}^{m} \|X(\hat{\beta}^{(k)} - \hat{\beta}^{(k-1)})\|_2^2;$$
2. after m steps, only m coordinates, or groups of coordinates, have been updated so we know that the solution is sparse and blocky, at least when m is small. In our experimental study, very often, only a few moves are necessary, i.e. $\|X(\hat{\beta}^{(m)} - \hat{\beta}^{(m-1)})\|_2^2$ becomes very small after only a few steps.

3.1 A Note on the Computational Complexity of our Method

Let us remark that we can easily upper bound the computational complexity of the "best single move" strategy with a limited number of steps M (like in Theorems 1 and 2).

First, we need to explore all the possible groups of variables, there are $p + (p-1) ... + (p-K+1) \leq Kp$ such groups. Then for each group we have to compute the quantities in (2), this costs roughly $\mathcal{O}(nk) = \mathcal{O}(nK)$ operations.

This means that the computation time is roughly $\mathcal{O}(npK^2M)$. So, of course, if we know that the blocks in β have a limited size, this is reasonable, but if we do not have such an information and we take $K = p$, we can have trouble in the case where p is large.

4 Simulations

4.1 Description of the Experiments

A toy example is proposed in the seminal paper of Tibshirani [18]. We slightly modify this example to have

$$\forall i \in \{1, \ldots, 50\}, \quad Y_i = \beta' X_i + \varepsilon_i$$

with $X_i \in \mathbb{R}^p$, $\beta \in \mathbb{R}^p$ and the ε_i are i. i. d. from a gaussian $\mathcal{N}(0, \sigma^2)$. The X_i's are i.i.d., and each X_i is drawn from a Gaussian distribution with mean $(0, \ldots, 0)'$ and with variance matrix:

$$\Sigma(\rho) = \left(\rho^{|i-j|}\right)_{\substack{i \in \{1, \ldots, p\} \\ j \in \{1, \ldots, p\}}}$$

for $\rho \in [0, 1[$. Note that Tibshirani's toy example is set with $p = 8$. Here we will consider $p > 8$.

We will use two values for β:

$$\beta^* = (5, 5, 5, 5, 5, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, ..., 0) \in \mathbb{R}^p,$$
$$\beta^{**} = (5, 4.5, 4, 3.5, 3, 2.5, 2, 1.5, 1, 0.5, 0, ..., 0) \in \mathbb{R}^p,$$

β^* is sparse and blocky while β^{**} is sparse and, in some way, smooth (a context that should be in favor of the S-LASSO estimator). We use two values for σ: 1 ("low noise") and 3 ("noisy case"); one value for ρ: $\rho = 0.5$, and finally two values for p: $p = 15 < n$ and $p = 100 > n$.

The S-LASSO and Fused-LASSO estimators are computed via the Pathwise Coordinate Optimization procedure described in [8]. Our procedure will be called in the experiments ISBF (Iterative Selection of Blocks of Features). It is used with the "best single move" strategy, and with $K = 10$.

We will finally use the parameters s and t in a grid:

$$s, t \in \mathcal{G} = \left\{ 10^{-i/8} \sqrt{\frac{\sigma^2 \log(p)}{n}}; \quad i = 0, ..., 20 \right\}. \tag{3}$$

4.2 Results

In a first time, we present in detail the results for one particular experiment. We then give an overview of the results in the whole set of experiments.

First, we focus on an experiment realized in the case $\beta = \beta^*$, $\sigma = 1$, $\rho = 0.5$ and $p = 15$. We define, for a given $s \in \mathcal{G}$, $L_{\mathrm{ISBF},s}$ as the loss obtained using the ISBF method with threshold value s, and

$$L_{\mathrm{F},s} = \arg\min_{t \in \mathcal{G}} L(\tilde{\beta}^F_{s,t})$$

the oracle for the Fused-LASSO with s fixed. We define in the same way $L_{\mathrm{S},s}$ for the S-LASSO. Figure 1 gives a plot of $L_{\mathrm{ISBF},s}$, $L_{\mathrm{F},s}$ and $L_{\mathrm{S},s}$ as a function of s.

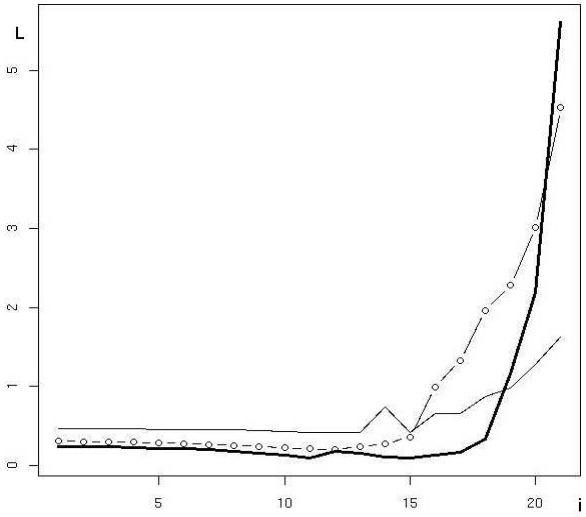

Fig. 1. The quantities $L_{\mathrm{ISBF},s}$ (thick line), $L_{\mathrm{F},s}$ (thin line) and $L_{\mathrm{S},s}$ (dotted line) as a function of s. The horizontal axis gives $i \in \{0, ..., 20\}$, the vertical axis is the value of the risk $L_{\mathrm{ISBF},s}$, $L_{\mathrm{F},s}$ and $L_{\mathrm{S},s}$ with $s = s(i)$ as defined in Equation 3.

Note that $L_{\mathrm{ISBF},s}$ is almost always under $L_{\mathrm{F},s}$ and $L_{\mathrm{S},s}$. So, for any reasonable s, the ISBF procedure reaches a better performance than the *oracle* of the Fused-LASSO and the S-LASSO - note that the oracles are not even available to the practitioners. In practice, we use some data-driven method to choose s for ISBF and (s, t) for the LASSO-type procedures, as cross-validation. Note that ISBF is more easy to deal with as it involves only one parameter to tune.

Now, let us have a look at the whole set of experiments (for each pair (σ, p) we run 20 experiments). For the Fused-LASSO and the S-LASSO, we report the

performance of the oracle: namely, for each simulation, we define

$$L_{\mathrm{F}} = \arg \min_{(s,t)\in\mathcal{G}^2} L(\tilde{\beta}^F_{s,t})$$

the oracle loss for the Fused-LASSO, and we do the same to define the oracle loss of the S-LASSO, L_{S}, and for the ISBF, L_{ISBF}. The results for the estimation of β^* are reported in Table 1.

Table 1. Results for the estimation of β^* (sparse and blocky).

σ	p		L_{F}	L_{S}	L_{ISBF}
1	15	median	0.41	0.18	**0.10**
		mean	0.52	0.24	**0.14**
		s.d.	0.28	0.15	0.11
3	15	median	0.49	0.70	**0.41**
		mean	0.75	0.71	**0.46**
		s.d.	0.58	0.39	0.28
1	100	median	0.51	0.55	**0.35**
		mean	0.64	0.55	**0.38**
		s.d.	0.32	0.13	0.16
3	100	median	0.92	1.32	**0.75**
		mean	0.95	1.25	**0.77**
		s.d.	0.46	0.29	0.32

We can see that ISBF clearly outperforms the other methods on this particular set of experiments. The results for the estimation of β^{**} are reported in Table 2. Here, the S-LASSO is better.

Table 2. Results for the estimation of β^{**} (sparse and smooth).

σ	p		L_{F}	L_{S}	L_{ISBF}
1	15	median	0.51	**0.12**	0.21
1	15	mean	0.61	**0.13**	0.24
		s.d.	0.52	0.05	0.08
3	15	median	0.85	**0.36**	0.63
		mean	0.91	**0.39**	0.76
		s.d.	0.37	0.28	0.44
1	100	median	0.44	**0.31**	0.66
		mean	0.47	**0.35**	0.64
		s.d.	0.12	0.12	0.08
3	100	median	0.81	**0.70**	1.21
		mean	1.08	**0.71**	1.22
		s.d.	0.80	0.22	0.22

A conclusion to the very short experimental study is that the IFSF seems better in the case of a sparse and blocky parameter, while the S-LASSO is better in the case of a sparse and smooth parameter.

5 Application on CGH Data

We now present an application to the detection of amplification or deletion of DNA using CGH arrays. The data we have are partial CGH arrays (only for chromosome 17) of cancer cells. The model (for one patient) is the following:

$$Y_i = \beta_i + \varepsilon_i$$

for $1 \leq i \leq n$ with $n = 7728$, here i represents ordered positions of measurements on the chromosome. Remark that this is a particular case of Model (1) with $p = n$ and $X = I_n$ (this context is the one studied by Rinaldo [16]), here i represents ordered positions of measurements on the chromosome of the patient.

The data we use are on a logarithmic scale, so, $\beta_i = 0$ means no amplification or deletion of DNA at position i, $\beta_i < 0$ means deletion and $\beta_i > 0$ means amplification. We use ISBF to estimate β, the results are given in Figures 2 and 3 (for two different patients, with $K = 100$ and $s \simeq 0.01$). Note that there is work left to compare this method to the already available ones in this context. As an illustration we also give here the results of the estimation using the S-LASSO procedure with $s = t = 0.4$, see Figures 4 and 5. However, it is a good point to see that our method works in a reasonable time on a large dataset and gives reasonable results, probably more suited for interpretation by a practitioner.

Remark 1 *All simulations and experiments were performed with the R software [14]. The code is available from the author.*

6 Proof of Theorem 1

We now give the proof of our main result. First, we give some preliminary definitions and result.

Definition 1 *For the sake of simplicity let us put* $K_j = \min(K, p - j + 1)$.

Actually K_j is the maximal length for a group $\{j, j+1, ..., j+k-1\}$ constrained by $j + k - 1 \leq p$ and $k \leq K$.

Definition 2 *Let us put, for any* $s > 0$, $j \in \{1, ..., p\}$ *and* $k \in \{1, ..., K_j\}$*:*

$$R_s(j,k) = \left\{ b \in \mathbb{R}^p : \left| \frac{\sum_{h=j}^{j+k-1} X_h'(Y - Xb)}{\sqrt{\mathbf{1}_{j,k}' X' X \mathbf{1}_{j,k}}} \right| \leq s \right\}$$

and, for any $s > 0$ *and* $b \in \mathbb{R}^p$ *let us define the event*

$$A_s(b) = \bigcap_{j=1}^{p} \bigcap_{k=1}^{K_j} \{ b \in R_s(j,k) \}.$$

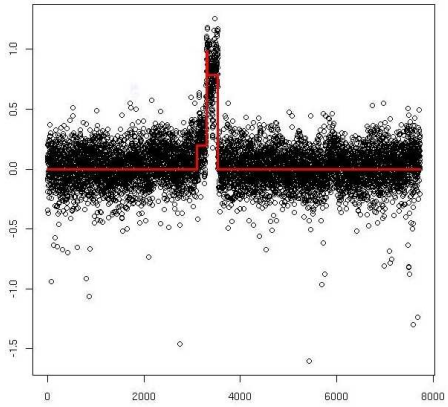

Fig. 2. The data Y (in black) and the estimated β (in red) using Iterative Selection of Blocks of Features, for Patient 1.

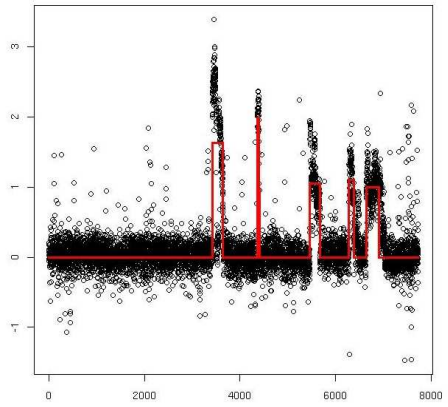

Fig. 3. The data Y (in black) and the estimated β (in red) using Iterative Selection of Blocks of Features, for Patient 2.

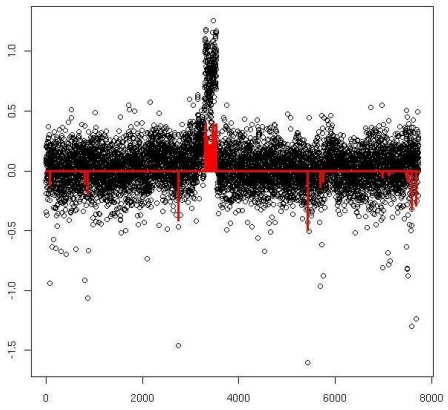

Fig. 4. The data Y (in black) and the estimated β (in red) using the S-LASSO estimator, for Patient 1.

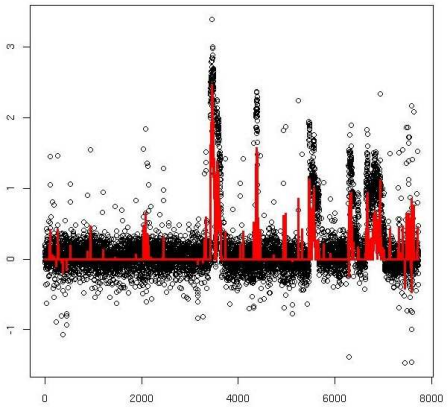

Fig. 5. The data Y (in black) and the estimated β (in red) using the S-LASSO estimator, for Patient 2.

Lemma 1 *For any $s > 0$ we have*

$$P\left[A_s(\beta)\right] \geq 1 - K\left(p - \frac{K-1}{2}\right) e^{-\frac{s^2}{2\sigma^2}}.$$

Proof. First, let us note that (1) implies that

$$X'(Y - X\beta) \sim \mathcal{N}(0, \sigma^2 X'X)$$

and so for any $j \in \{1, ..., p\}$ and $k \in \{1, ..., K_j\}$,

$$\sum_{h=j}^{j+k-1} X_h'(Y - X\beta) \sim \mathcal{N}(0, \sigma^2 \mathbf{1}_{j,k}' X' X \mathbf{1}_{j,k})$$

or

$$\frac{\sum_{h=j}^{j+k-1} X_h'(Y - X\beta)}{\sqrt{\mathbf{1}_{j,k}' X' X \mathbf{1}_{j,k}}} \sim \mathcal{N}(0, \sigma^2).$$

This implies that, for any $j \in \{1, ..., p\}$, for any $k \in \{1, ..., K_j\}$ and for any $s > 0$,

$$P\left(\frac{|X_h'(Y - X\beta)|}{\sqrt{\mathbf{1}_{j,k}' X' X \mathbf{1}_{j,k}}} > s \right) \leq e^{-\frac{s^2}{2\sigma^2}}$$

and, by a union bound argument over all $j \in \{1, ..., p\}$ and $k \in \{1, ..., K_j\}$,

$$P\left(\exists j \in \{1, ..., p\}, \exists k \in \{1, ..., K_j\} : \quad \frac{\left|\sum_{h=j}^{j+k-1} X_h'(Y - X\beta)\right|}{\sqrt{\mathbf{1}_{j,k}' X' X \mathbf{1}_{j,k}}} > s \right)$$
$$\leq e^{-\frac{s^2}{2\sigma^2}} \sum_{\ell=0}^{K-1} (p - \ell) = \left[Kp - \frac{K(K-1)}{2} \right] e^{-\frac{s^2}{2\sigma^2}}.$$

This ends the proof. ■

Definition 3 *For any closed and convex set $\mathcal{C} \subset \mathbb{R}^p$ we define $\Pi_C(.)$ the orthogonal projection on $\mathcal{C}$ with respect to the norm induced by X:*

$$\Pi_C(b) = \arg\min_{p \in \mathcal{C}} \|X(b - p)\|_2^2.$$

We are now ready to give the proof of Theorem 1. The proof heavily uses the geometrical considerations given in [2].

Proof of Theorem 1. From now, let us fix $s > 0$ and assume that we are in the event $A_s(\beta)$ (according to Lemma 1, this is true with probability at least $1 - pK \exp(-s^2/(2\sigma^2))$).

So, for any $j \in \{1, ..., p\}$, $k \in \{1, ..., K_j\}$, we have $\beta \in R_s(j, k)$. Moreover, note that $R_s(j, k)$ is convex and closed. Using classical convex analysis result (see [2] for example), we have, for any $b \in \mathbb{R}^p$,

$$\|X(\Pi_{R_s(j,k)}(b) - \beta)\|_2^2 \leq \|X(b - \beta)\|_2^2 - \|X(\Pi_{R_s(j,k)}(b) - b)\|_2^2, \tag{4}$$

see Figure 6 for an illustration.

The last point of the proof will be to remark that, once that j and k are fixed, the $\hat{\beta}^{(m+1)}$ defined in the "1 coordinate move" satisfies

$$\hat{\beta}^{(m+1)} = \Pi_{R_s(j,k)}(\hat{\beta}^{(m)}).$$

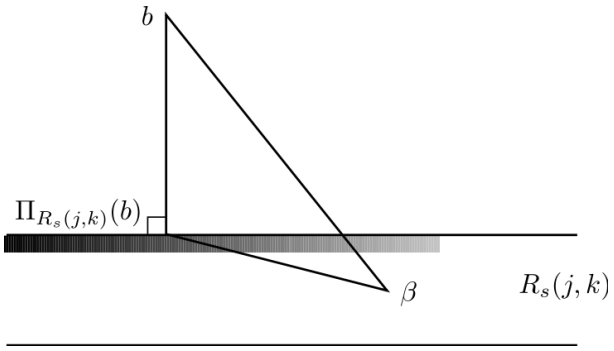

Fig. 6. Illustration of Equation 4.

To see this, remark that the projection $\mathit{\Pi}_{R_s(j,k)}(\hat{\beta}^{(m)})$ is defined by the program

$$\begin{cases} \min_{u\in\mathbb{R}^p} \|X(\hat{\beta}^{(m)}-u)\|_2^2 \\ s.t. \left|\frac{\sum_{h=j}^{j+k-1} X_h'(Y-Xu)}{\sqrt{\mathbf{1}_{j,k}' X'X\mathbf{1}_{j,k}}}\right| \leq s, \end{cases}$$

of which the solution is given by (2). ■

7 Conclusion and Perspectives

In this paper, we tried to give an estimator that takes advantage of the sparsity of β as well as of its blockwise aspect in the setting of regression in large dimension.

We proposed a very simple algorithm that performs well in practice, and give directions for the study of its theoretical performances.

In future works, we would like to give more precise theoretical results - in the spirit of [3] on the LASSO.

Eventually, we would like to point out some similarities between our estimator and other well-known estimators. In the proof of our main result, we presented our sequence of estimators $(\hat{\beta}^{(m)})_m$ as successive projections on various confidence regions of the form

$$R_s(j,k) = \left\{ b\in\mathbb{R}^p : \left|\frac{\sum_{h=j}^{j+k-1} X_h'(Y-Xb)}{\sqrt{\mathbf{1}_{j,k}' X'X\mathbf{1}_{j,k}}}\right| \leq s \right\}.$$

We may wonder what happens if we project directly the first value $\hat{\beta}^{(0)}=0$ into the intersections of these confidence regions. We obtain an estimator $\hat{\beta}(s,t)$

defined by the following program:

$$\hat{\beta}(s,t) = \begin{cases} \arg\min_{b\in\mathbb{R}^p} \|Xb\|_2^2 \\ \\ s.t. \sup_{j,k} \left| \frac{\sum_{h=j}^{j+k-1} X_h'(Y-Xb)}{\sqrt{\mathbf{1}_{j,k}' X'X \mathbf{1}_{j,k}}} \right| \leq s. \end{cases} \tag{5}$$

If we fix $K = 1$ (so we only work with constraints defined by single variables and not by groups of variables), we obtain

$$\begin{cases} \min_{b\in\mathbb{R}^p} \|Xb\|_2^2 \\ \\ s.t. \quad \|\frac{1}{n} X'(Y - Xb)\|_\infty \leq s \end{cases}$$

that was proved to have as a solution the LASSO estimator given by

$$\min_{b\in\mathbb{R}^p} \left\{ \|Y - Xb\|_2^2 + 2ns \sum_{j=1}^{p} |b_j| \right\},$$

see for example [13] for the proof, and [2] for a discussion of the geometric role of the constraint $\|(1/n)X'(Y - Xb)\|_\infty \leq s$. So, in some way, the estimator $\hat{\beta}(s,t)$ is really a variant of the LASSO, that tries to take into account the fact that β has blocks. The study of $\hat{\beta}(s,t)$ will be the object of future works.

Acknowledgments We would like to thank Dr. Jean-Paul Feugeas (INSERM U944, Institut Universitaire d'Hématologie) who provided the CGH data with detailed explanations. We would also like to thank Dr. Mohamed Hebiri (ETH - Zürich) and Joseph Salmon (Université Paris 7) as well as the anonymous referees for useful comments. The project was funded by: ANR PARCIMONIE.

References

1. Alquier, P. Iterative feature selection in regression estimation. *Annales de l'Institut Henri Poincaré, Probability and Statistics 44*, 1 (2008), 47–88.
2. Alquier, P. LASSO, iterative feature selection and the correlation selector: Oracle inequalities and numerical performances. *Electron. J. Stat.* (2008), 1129–1152.
3. Bickel, P., Ritov, Y., and Tsybakov, A. Simultaneous analysis of LASSO and dantzig selector. *The Annals of Statistics 37*, 4 (2009), 1705–1732.
4. Breiman, L. Better subset regression using the nonnegative garrote. *Technometrics 37* (1995), 373–384.
5. Bunea, F., Tsybakov, A., and Wegkamp, M. Sparsity oracle inequalities for the lasso. *Electron. J. Stat. 1* (2007), 169–194.
6. Candès, E., and Tao, T. The dantzig selector: statistical estimation when p is much larger than n. *Ann. Statist. 35* (2007).
7. Chesnau, C., and Hebiri, M. Some theoretical results on the grouped variables lasso. *Mathematical Methods of Statistics 17*, 4 (2008), 317–326.

8. Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. Pathwise coordinate optimization. *Ann. Appl. Statist. 1*, 2 (2007), 302–332.
9. Hebiri, M. Regularization with the smooth-LASSO procedure. Preprint LPMA, arXiv:0803.0668, 2008.
10. Hebiri, M., and Van de Geer, S. The smooth-lasso and other $\ell_1 + \ell_2$-penalized methods. arXiv:1003.4885, 2010.
11. Hoefling, H. A path algorithm for the fused LASSO signal approwimator. Preprint arXiv:0910.0526, 2009.
12. Huang, J., Salim, A., Lei, K., O'Sullivan, K., and Pawitan, Y. Classification of array cgh data using smoothed logistic regression model. *Statistics in Medicine 8*, 30 (2009), 3798–3810.
13. Osborne, M., Presnell, B., and Turlach, B. On the LASSO and its dual. *J. Comput. Graph. Statist. 9*, 2 (2000), 319–337.
14. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.
15. Rapaport, F., Barillot, E., and Vert, J.-P. Classification of array-CGH data using fused SVM. *Bioinformatics 24*, 13 (2008), 1375,1382.
16. Rinaldo, A. Properties and refinements of the fused LASSO. *The Annals of Statistics 37*, 5B (2009), 2922–2952.
17. Slawski, M., zu Castell, W., and Tutz, G. Feature selection guided by structural information. To appear in the Annals of Applied Statistics.
18. Tibshirani, R. Regression shrinkage and selection via the LASSO. *J. Roy. Statist. Soc. Ser. B 58*, 1 (1996), 267–288.
19. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. Sparsity and smoothness via the fused lasso. *JRSS-B 67*, 1 (2005), 91–108.
20. Tibshirani, R. J., and Taylor, J. Regularization path for least squares problems with generalized ℓ_1 penalties. Preprint, 2009.
21. Van de Geer, S., and Bühlmann, P. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics 3* (2009), 1360–1392.
22. Yuan, M., and Lin, Y. Model selection and estimation in regression with grouped variables. *JRSS-B 68*, 1 (2006), 49–67.
23. Zhao, P., Rocha, G., and Yu, B. The composite absolute penalties for grouped and hierarchical variable selection. *The Annals of Statistics 37*, 6A (2009), 3468–3497.