ROBUST SPARSITY AND CLUSTERING REGULARIZATION FOR REGRESSION

Xiangrong Zeng and Mário A. T. Figueiredo

Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal

ABSTRACT

Based on our previously proposed *SPARsity and Clustering* (SPARC) regularization, we propose a robust variant of SPARC (RSPARC), which is able to detect observations corrupted by sparse outliers. The proposed RSPARC inherits the ability of SPARC to promote group-sparsity, and combines that ability with robustness to outliers. We propose algorithms of the alternating direction method of multipliers (ADMM) family to solve several regularization formulations involving SPARC regularization. Experiments show that RSPARC is a competitive robust group-sparsity-inducing regularization for regression.

Index Terms— Sparsity and clustering, group sparsity, Lasso, elastic net.

1. INTRODUCTION

Consider the standard linear regression model, where the goal is to estimate a vector of regression coefficients $\mathbf{x} \in \mathbb{R}^n$, from a vector of responses $\mathbf{y} \in \mathbb{R}^m$, given by

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the so-called design matrix (which is known) and $\mathbf{e} \in \mathbb{R}^m$ is the additive measurement noise. In most cases, the solution of this type of problem (which is at the core of statistics, machine learning, and signal processing) requires *regularization*, that is, the mathematical specification of properties that an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ should (or is known to) satisfy, in addition to providing a good explanation to the observed responses $\mathbf{y}$.

There are three standard formulations, depending on how the regularizer $\phi(\mathbf{x})$ and the data-fidelity term $f(\mathbf{x})$ are combined to achieve a balance between the two goals [1]:

1. Tikhonov regularization:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \tau\, \phi(\mathbf{x}), \tag{2}$$

2. Morozov regularization:

$$\min_{\mathbf{x}} \phi(\mathbf{x}) \;\text{ s.t. } f(\mathbf{x}) \leq \varepsilon \tag{3}$$

3. Ivanov regularization:

$$\min_{\mathbf{x}} f(\mathbf{x}) \;\text{ s.t. } \phi(\mathbf{x}) \leq \epsilon. \tag{4}$$

Typically, $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ is the data-fidelity term (under a white Gaussian noise assumption) and $\phi(\mathbf{x})$ is the regularizer that enforces certain properties on the target solution; finally, τ, ε, and ϵ are non-negative parameters.

In the past decade, not only in signal processing (mainly due to the advent of compressing sensing), but also in statistics and machine learning, a significant amount of work has been devoted to regularizers that encourage sparse solutions (*e.g.*, the famous sparsity-promoting LASSO regularizer $\phi_{\text{LASSO}}(\mathbf{x}) = \|\mathbf{x}\|_1$). More recently, much attention has been focused, not only on simple sparsity, but also on structured/group sparsity [2], with the appearance of several group-sparsity regularizers: *group LASSO* (gLASSO) [3], *fused LASSO* (fLASSO) [4], *elastic net* (EN) [5], *octagonal shrinkage and clustering algorithm for regression* (OSCAR) [6], and several others, not listed here due to space limitations (see review in [2]). However, gLASSO (and its many variants [2]) requires prior knowledge about the group structure, which is a strong requirement in many applications, while fLASSO depends on a given order of variables; these two classes of approaches are thus better suited to signal processing applications than to variable selection and grouping in machine learning problems, such as regression or classification (where the order of the variables is often meaningless). In contrast, both EN and OSCAR were proposed for regression problems and do not rely on any ordering of the variables or knowledge about group structure. The EN regularizer is defined as

$$\phi_{\text{EN}}^{\lambda_1,\lambda_2}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2\|\mathbf{x}\|_2^2,$$

and the OSCAR regularizer (shown in [7] to outperform EN in feature grouping) is defined as

$$\phi_{\text{OSCAR}}^{\lambda_1,\lambda_2}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i<j}\max\{|x_i|,|x_j|\},$$

where λ_1 and λ_2 are non-negative parameters [7]. The ℓ_1 norm and the pairwise ℓ_∞ penalty in OSCAR simultaneously encourage the components to be sparse and equal in magnitude, respectively. However, it may happen that components with small magnitude that should be shrunk to zero by the ℓ_1 norm are also penalized by the pairwise ℓ_∞ term, which may prevent accurate grouping; moreover, components with large magnitude that should simply be grouped by the pairwise ℓ_∞ norm are also shrunk by the ℓ_1 norm (see Figure 1). To overcome these drawbacks, we have proposed the *SPARsity-and-Clustering* (SPARC) regularizer [8], [9], where the pairwise ℓ_∞ penalty is applied only to the non-zero elements. The SPARC regularizer is given by

$$\phi_{\text{SPARC}}^{\lambda,K}(\mathbf{x}) = \iota_{\Sigma_K}(\mathbf{x}) + \lambda\sum_{i,j\in\Omega_K(\mathbf{x}),\, i<j}\max\{|x_i|,|x_j|\}, \tag{5}$$

where ι_C denotes the indicator of set C (*i.e.*, $\iota_C(\mathbf{x}) = 0$, if $\mathbf{x} \in C$; $\iota_C(\mathbf{x}) = +\infty$, if $\mathbf{x} \notin C$), $\Sigma_K = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq K\}$ is the set of K-sparse vectors, and $\Omega_K(\mathbf{x}) = \text{supp}(\mathcal{P}_{\Sigma_K}(\mathbf{x}))$ (where $\mathcal{P}_{\Sigma_K}(\mathbf{x})$ is the projection on Σ_K, and $\text{supp}(\mathbf{v}) = \{i : v_i \neq 0\}$) is the set

This work was partially supported by Fundao para a Ciłncia e Tecnologia, grants PEst-OE/EEI/LA0008/2013 and PTDC/EEI-PRO/1470/2012. Xiangrong Zeng is partially supported by grant SFRH/BD/75991/2011.

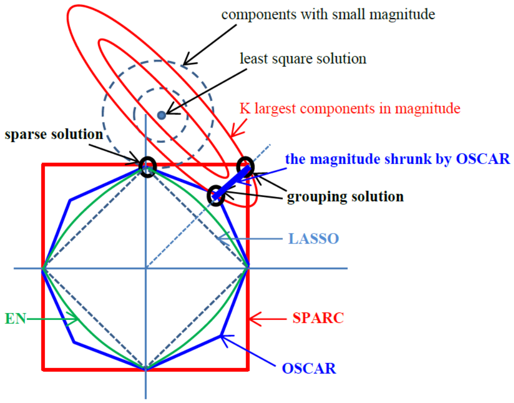

components with small magnitude
least square solution
K largest components in magnitude
sparse solution
the magnitude shrunk by OSCAR
grouping solution
LASSO
EN
SPARC
OSCAR

Fig. 1. Demonstration of different regularizers

of indices of the K largest components of $\mathbf{x}$ (in magnitude). This regularizer enforces K-sparsity and encourages the non-zeros to be equal in magnitude.

However, the regularization schemes (2), (3), or (4), with any of the above regularizers, are not able to address the case where the response vector $\mathbf{y}$ is contaminated by outliers, which usually occur infrequently and hence are sparse. One way that has been proposed to deal with outliers is to add a sparse variable $\mathbf{w}$ to (1) [10], [11], [12], [13], [14], [15], yielding

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} + \mathbf{w}. \tag{6}$$

If y_i is not an outlier, then $w_i = 0$, whereas if y_i is an outlier, then w_i can be viewed as the anomalous error. Following [13] and [15], a robust variant of LASSO (referred to as RLASSO) can be defined as

$$\min_{\mathbf{x},\mathbf{w}} \tfrac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{w}\|_2^2 + \tau\|\mathbf{w}\|_1 + \lambda\|\mathbf{x}\|_1 \tag{7}$$

where τ and λ are non-negative parameters.

To the best of our knowledge, there are no methods that simultaneously encourage group sparsity and adaptively detect outliers. In this paper, we propose a robust variant of our previously proposed SPARC, which combines the group-sparsity-inducing capability of SPARC and outlier-detection technique of RLASSO. We propose algorithms of the ADMM family [16], more specifically variants of the CSALSA algorithm [17], to solve several regularization formulations involving SPARC regularization and sparse outlier detection, and illustrate the performance of the proposed method on a benchmark regression problem.

2. PROPOSED FORMULATION AND APPROACH

2.1. Proposed Formulation

A robust version of SPARC (RSPARC) can be formulated using the three regularization schemes mentioned above:

1. Tikhonov regularization (referred to as RSPARC-T)

$$\min_{\mathbf{x},\mathbf{w}} \tfrac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{w}\|_2^2 + \tau_1\,\phi^{\lambda,K}_{\text{SPARC}}(\mathbf{x}) + \tau_2\|\mathbf{w}\|_1 \tag{8}$$

2. Morozov regularization (referred to as RSPARC-M)

$$\min_{\mathbf{x},\mathbf{w}} \|\mathbf{w}\|_1 + \varepsilon_1\phi^{\lambda,K}_{\text{SPARC}}(\mathbf{x}), \;\; \text{s.t. } \|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{w}\|_2 \le \varepsilon_2 \tag{9}$$

3. Ivanov regularization (referred to as RSPARC-I)

$$\min_{\mathbf{x},\mathbf{w}} \|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{w}\|_2^2, \;\; \text{s.t. } \phi^{\lambda,K}_{\text{SPARC}}(\mathbf{x}) \le \epsilon_1, \|\mathbf{w}\|_1 \le \epsilon_2, \tag{10}$$

where $\tau_1, \tau_2, \varepsilon_1, \varepsilon_2, \epsilon_1$ and ϵ_2 are non-negative parameters.

2.2. Key Ingredients

2.2.1. Proximity operator of SPARC regularizer

A key ingredient for algorithms to solve the problems (8) and (9) is the proximity operator

$$\text{prox}_{\phi^{\lambda,K}_{\text{SPARC}}}(\mathbf{v}) = \arg\min_{\mathbf{x}} \left(\phi^{\lambda,K}_{\text{SPARC}}(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2\right). \tag{11}$$

The key observation that allows computing $\text{prox}_{\phi^{\lambda,K}_{\text{SPARC}}}(\mathbf{v})$ is

$$\mathbf{v}\in\Sigma_K \;\Rightarrow\; \phi^{\lambda,K}_{\text{SPARC}}(\mathbf{v}) = \phi^{0,\lambda}_{\text{OSCAR}}\left(\mathbf{v}_{\Omega_K(\mathbf{v})}\right), \tag{12}$$

where $\mathbf{v}_S \in \mathbb{R}^{|S|}$ is the sub-vector of $\mathbf{v}$ indexed by an index subset $S \subseteq \{1,...,p\}$. Combining this with properties of proximity operators and ideas from [18] allows showing (details are omitted here, for lack of space) that $\mathbf{z} = \text{prox}_{\phi^{\lambda,K}_{\text{SPARC}}}(\mathbf{v})$ can be computed as follows:

$$\mathbf{z}_{\Omega_K(\mathbf{v})} = \text{prox}_{\phi^{0,\lambda}_{\text{OSCAR}}}(\mathbf{v}_{\Omega_K(\mathbf{v})}), \qquad \mathbf{z}_{\overline{\Omega}_K(\mathbf{v})} = \mathbf{0} \tag{13}$$

where $\mathbf{0}$ is a vector of zeros, $\overline{\Omega}_K(\mathbf{v}) = \{1,...,p\}\setminus\Omega_K(\mathbf{v})$, and $\text{prox}_{\phi^{0,\lambda}_{\text{OSCAR}}}$ can be calculated exactly or approximately by the *grouping proximity operator* and the *approximate proximity operator*, as proposed in [19].

2.2.2. Projection onto SPARC ball

A central building block for solving (10) is the projection onto an ϵ-radius SPARC ball,

$$\mathcal{C}^{\lambda,K}_{\epsilon} = \left\{\mathbf{x} : \phi^{\lambda,K}_{\text{SPARC}}(\mathbf{x}) \le \epsilon\right\}, \tag{14}$$

which, by the definition of Euclidean projection, is given by

$$\text{proj}_{\mathcal{C}^{\lambda,K}_{\epsilon}}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathcal{C}^{\lambda,K}_{\epsilon}} \|\mathbf{v}-\mathbf{x}\|_2^2. \tag{15}$$

In the same vein as the previous sub-section, we can also compute this projection using that of the OSCAR ball, which was addressed in our previous work [20]. Let the ϵ-radius OSCAR ball be denoted by

$$\mathcal{R}^{\lambda_1,\lambda_2}_{\epsilon} = \left\{\mathbf{x} : \phi^{\lambda_1,\lambda_2}_{\text{OSCAR}}(\mathbf{x}) \le \epsilon\right\}.$$

According to (12), $\mathbf{u} = \text{proj}_{\mathcal{C}^{\lambda,K}_{\epsilon}}(\mathbf{v})$ can be computed as follows:

$$\mathbf{u}_{\Omega_K(\mathbf{v})} = \text{proj}_{\mathcal{R}^{\lambda_1,\lambda_2}_{\epsilon}}(\mathbf{v}_{\Omega_K(\mathbf{v})}), \qquad \mathbf{u}_{\overline{\Omega}_K(\mathbf{v})} = \mathbf{0} \tag{16}$$

where

$$\text{proj}_{\mathcal{R}^{\lambda_1,\lambda_2}_{\epsilon}}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathcal{R}^{\lambda_1,\lambda_2}_{\epsilon}} \|\mathbf{v}-\mathbf{x}\|_2^2 \tag{17}$$

is a simply special case of the problems, with the sorted ℓ_1 ball constraint and a convex and continuously differentiable objective function, as addressed in [20], since $\mathcal{R}^{\lambda_1,\lambda_2}_{\epsilon}$ is a simply special instance of the sorted ℓ_1 ball and here the objective is $\|\mathbf{v}-\mathbf{x}\|_2^2$.

2.3. Proposed Algorithms

Naturally, solving (8), (9), and (10) can be addressed by alternating minimization w.r.t $\mathbf{x}$ and $\mathbf{w}$. In this section, we adopt *constrained split augmented Lagrangian shrinkage algorithm* (CSALSA) [17] to solve subproblems in the forms of (8), (9) and (10), termed CSALSA-T, CSALSA-M and CSALSA-I, respectively.

2.3.1. Solving RSPARC-T

An algorithmic framework to solve (8) is given as follows.

Algorithm *Framework to solve RSPARC-T*

```
1.  Input $\mathbf{y}, \mathbf{A}, K, \lambda, \tau_1, \tau_2$ and $\mathbf{w}_0 = \mathbf{0}$.
2.  $k = 1$
3.  repeat
4.      $\mathbf{x}_{k+1} = \text{CSALSA-T}_{\mathbf{x}}\left(\mathbf{y} - \mathbf{w}_k, \mathbf{A}, K, \lambda, \tau_1\right)$
5.      $\mathbf{w}_{k+1} = \text{soft}\left(\mathbf{y} - \mathbf{A}\mathbf{x}_{k+1}, \tau_2\right)$
6.      $k \leftarrow k + 1$
7.  until some stopping criterion is satisfied.
8.  Output $\mathbf{x}_k$ and $\mathbf{w}_k$.
```

where "soft" is the well-known soft thresholding:

$$\text{soft}(\mathbf{z}, \tau) = \text{sign}(\mathbf{z}) \odot \max\left\{|\mathbf{z}| - \tau, 0\right\}.$$

In the algorithm above, CSALSA-T$_{\mathbf{x}}$ denotes the application of the CSALSA algorithm to solve the subproblem (8) w.r.t $\mathbf{x}$, as follows:

Algorithm *CSALSA-T*$_{\mathbf{x}}(\mathbf{z}, \mathbf{A}, K, \lambda, \tau_1)$

```
1.  Set $k = 0$, $\alpha > 0$, $\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, \mathbf{d}_0^{(1)}, \mathbf{d}_0^{(2)}$.
2.  repeat
3.      $\mathbf{r}_k = \mathbf{A}^T\left(\mathbf{x}_0^{(1)} + \mathbf{d}_0^{(1)}\right) + \mathbf{x}_0^{(2)} + \mathbf{d}_0^{(2)}$
4.      $\mathbf{u}_{k+1} = \left(\mathbf{A}^T\mathbf{A} + \mathbf{I}\right)^{-1}\mathbf{r}_k$
5.      $\mathbf{x}_{k+1}^{(1)} = \text{prox}_{\frac{1}{2}\|\cdot - \mathbf{z}\|_2^2/\alpha}\left(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{d}_k^{(1)}\right)$
6.      $\mathbf{x}_{k+1}^{(2)} = \text{prox}_{\tau_1\phi_{\text{SPARC}}^{\lambda,K}/\alpha}\left(\mathbf{u}_{k+1} - \mathbf{d}_k^{(2)}\right)$
7.      $\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - \mathbf{A}\mathbf{u}_{k+1} + \mathbf{x}_{k+1}^{(1)}$
8.      $\mathbf{d}_{k+1}^{(2)} = \mathbf{d}_k^{(2)} - \mathbf{u}_{k+1} + \mathbf{x}_{k+1}^{(2)}$
9.      $k \leftarrow k + 1$
10. until some stopping criterion is satisfied.
11. return $\mathbf{x}_k^{(1)}$.
```

Line 5 of this algorithm computes the proximity operator of the function $g(\mathbf{x}) = \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{z}\|_2^2$, which has a simple closed-form solution

$$\mathbf{x}_{k+1}^{(1)} = \tfrac{1}{1+\alpha}\left[\mathbf{z} + \alpha\left(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{d}^{(1)}\right)\right],$$

while line 6 is given by (13).

2.3.2. Solving RSPARC-M

The subproblems of alternatively minimizing (9) w.r.t $\mathbf{x}$ and $\mathbf{w}$ are solved by CSALSA-M$_{\mathbf{x}}$ and CSALSA-M$_{\mathbf{w}}$, respectively. Before proceeding, let us define

$$\mathcal{S}_\varepsilon^{\mathbf{z}} = \{\mathbf{x} : \|\mathbf{x} - \mathbf{z}\|_2 \leq \varepsilon\}, \tag{18}$$

as the ε-radius Euclidean ball centered at $\mathbf{z}$; then, the projecting of some $\mathbf{v}$ onto $\mathcal{S}_\varepsilon^{\mathbf{z}}$ is given by

$$\text{proj}_{\mathcal{S}_\varepsilon^{\mathbf{z}}}(\mathbf{v}) = \mathbf{z} + \begin{cases} \varepsilon\,\frac{\mathbf{v} - \mathbf{z}}{\|\mathbf{v} - \mathbf{z}\|_2}, & \text{if } \|\mathbf{v} - \mathbf{z}\|_2 > \varepsilon, \\ \mathbf{v} - \mathbf{z}, & \text{if } \|\mathbf{v} - \mathbf{z}\|_2 \leq \varepsilon. \end{cases} \tag{19}$$

The proposed algorithmic framework is given by

Algorithm *Framework to solve RSPARC-M*

```
1.  Input $\mathbf{y}, \mathbf{A}, K, \lambda, \varepsilon_1, \varepsilon_2$ and $\mathbf{w}_0 = \mathbf{0}$.
2.  $k = 1$
3.  repeat
4.      $\mathbf{x}_{k+1} = \text{CSALSA-M}_{\mathbf{x}}\left(\mathbf{y} - \mathbf{w}_k, \mathbf{A}, K, \lambda, \varepsilon_1, \varepsilon_2\right)$
5.      $\mathbf{w}_{k+1} = \text{CSALSA-M}_{\mathbf{w}}\left(\mathbf{y} - \mathbf{A}\mathbf{x}_{k+1}, \varepsilon_1\right)$
6.      $k \leftarrow k + 1$
7.  until some stopping criterion is satisfied.
8.  Output $\mathbf{x}_k$ and $\mathbf{w}_k$.
```

where CSALSA-M$_{\mathbf{x}}$ is very similar to CSALSA-T$_{\mathbf{x}}$, and amounts to replacing τ_1 of the latter by ε_1 and replacing line 5 of the latter by

$$\mathbf{x}_{k+1}^{(1)} = \text{proj}_{\mathcal{S}_{\varepsilon_2}^{\mathbf{z}}}\left(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{d}_k^{(1)}\right),$$

which can be computed by (19). Function CSALSA-M$_{\mathbf{w}}$ denotes the application of the CSALSA algorithm to solve a LASSO problem in the form of Morozov regularization w.r.t $\mathbf{w}$, as follows:

Algorithm *CSALSA-M*$_{\mathbf{w}}(\mathbf{z}, \varepsilon_2)$

```
1.  Set $k = 0$, $\alpha > 0$, $\mathbf{w}_0^{(1)}, \mathbf{w}_0^{(2)}, \mathbf{d}_0^{(1)}, \mathbf{d}_0^{(2)}$.
2.  repeat
3.      $\mathbf{u}_{k+1} = \frac{1}{2}\left(\mathbf{x}_0^{(1)} + \mathbf{d}_0^{(1)} + \mathbf{x}_0^{(2)} + \mathbf{d}_0^{(2)}\right)$
4.      $\mathbf{w}_{k+1}^{(1)} = \text{proj}_{\mathcal{S}_{\varepsilon_2}^{\mathbf{z}}}\left(\mathbf{u}_{k+1} - \mathbf{d}_k^{(1)}\right)$
5.      $\mathbf{w}_{k+1}^{(2)} = \text{soft}\left(\mathbf{u}_{k+1} - \mathbf{d}_k^{(2)}, 1/\alpha\right)$
6.      $\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - \mathbf{u}_{k+1} + \mathbf{w}_{k+1}^{(1)}$
7.      $\mathbf{d}_{k+1}^{(2)} = \mathbf{d}_k^{(2)} - \mathbf{u}_{k+1} + \mathbf{w}_{k+1}^{(2)}$
8.      $k \leftarrow k + 1$
9.  until some stopping criterion is satisfied.
10. return $\mathbf{x}_k^{(1)}$.
```

Note how the CSALSA algorithm avoids the (hard) projection on the ellipsoid in the constraint and uses a (simple) projection on an Euclidean ball, thanks to the use of variable splitting [17].

2.3.3. Solving RSPARC-I

Recall that the ϵ-radius SPARC ball is denoted as $\mathcal{C}_\epsilon^{\lambda,K}$ (see (14)), and let the ϵ-radius ℓ_1 ball be denoted by

$$\mathcal{B}_\epsilon = \left\{\mathbf{x} : \|\mathbf{x}\|_1 \leq \epsilon\right\}.$$

The projection onto $\mathcal{B}_\epsilon$ ($\text{proj}_{\mathcal{B}_{\epsilon_2}}$) is a well studied problem for which there are efficient algorithms [21], [22], [23].

Algorithm *Framework to solve RSPARC-I*

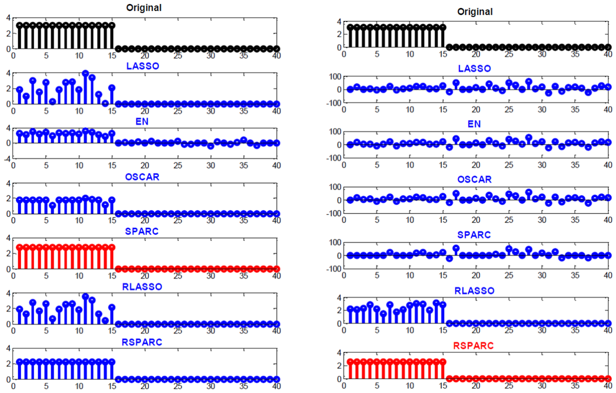
```
1.  Input $\mathbf{y}, \mathbf{A}, K, \lambda, \epsilon_1, \epsilon_2$ and $\mathbf{w}_0 = \mathbf{0}$.
2.  $k = 1$
3.  repeat
4.      $\mathbf{x}_{k+1} = \text{CSALSA-I}_{\mathbf{x}}\left(\mathbf{y} - \mathbf{w}_k, \mathbf{A}, K, \lambda, \epsilon_1\right)$
5.      $\mathbf{w}_{k+1} = \text{proj}_{\mathcal{B}_{\epsilon_2}}\left(\mathbf{y} - \mathbf{A}\mathbf{x}_{k+1}\right)$
6.      $k \leftarrow k + 1$
7.  until some stopping criterion is satisfied.
8.  Output $\mathbf{x}_k$ and $\mathbf{w}_k$.
```

where CSALSA-I$_{\mathbf{x}}$ is equivalent to CSALSA-T$_{\mathbf{x}}$, with line 6 replaced by

$$\mathbf{x}_{k+1}^{(2)} = \text{proj}_{\mathcal{C}_{\epsilon_1}^{\lambda,K}}\left(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{d}_k^{(2)}\right).$$

which can be computed by (16).

Note that, instead of CSALSA, we can use other state-of-the-art algorithms such as FISTA [24], TwIST [25], or SpaRSA [26].



(a) No corruption

(b) 5% corruption

Fig. 2. Obtained parameters by different regularizations

3. EXPERIMENTS

In this section, we report experiments aimed at comparing the proposed RSPARC with the LASSO, EN, OSCAR, SPARC, and, especially, RLASSO, all solved by the CSALSA algorithm [17]. Due to space limitation, we only focus on RSPARC-T (see (8), note that RSPARC-T, RSPARC-M and RSPARC-I are equivalent under mild conditions). We employ the following three metrics defined on an estimate $\hat{\mathbf{x}}$ of an original vector $\mathbf{x}$:

- Mean absolute prediction error: **MAPE** $= \left\|\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}})\right\|_1 /n$;
- Mean squared prediction error: **MSPE** $= \left\|\mathbf{A}(\mathbf{x}-\hat{\mathbf{x}})\right\|_2^2 /n$;
- Selection error rate: **SER** $= \left\||\mathrm{sign}(\mathbf{x})| - |\mathrm{sign}(\hat{\mathbf{x}})|\right\|_0 /n$;
- Degrees of freedom (**DoF**): the number of unique non-zero values that the coefficients of $\hat{\mathbf{x}}$ take (in comparison with the DoF of $\mathbf{x}$).

We consider a regression problem (also used in [7]) of the form (1), where the true parameter vector (which has DoF = 1) is

$$\mathbf{x} = [\underbrace{3,\cdots,3}_{15},\underbrace{0,\cdots,0}_{25}]^T \tag{20}$$

and the design matrix $\mathbf{A}$ is generated as follows:

$$\begin{aligned}
\mathbf{a}_i &= \mathbf{z}_1 + \epsilon_i^{\mathbf{x}},\ \mathbf{z}_1 \sim \mathcal{N}(0,1), i = 1,\ldots,5;\\
\mathbf{a}_i &= \mathbf{z}_2 + \epsilon_i^{\mathbf{x}},\ \mathbf{z}_2 \sim \mathcal{N}(0,1), i = 6,\ldots,10;\\
\mathbf{a}_i &= \mathbf{z}_3 + \epsilon_i^{\mathbf{x}},\ \mathbf{z}_3 \sim \mathcal{N}(0,1), i = 11,\ldots,15;\\
\mathbf{a}_i &\sim \mathcal{N}(0,1), i = 16,\ldots,40,
\end{aligned}$$

where $\epsilon_i^{\mathbf{x}}$ are i.i.d. $\mathcal{N}(0, 0.16)$, and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{40}]^T$ is further normalized. Finally, the noise $\mathbf{e}$ is i.i.d. $\mathcal{N}(0, 0.01)$.

Let r $(0 \leq r < 1)$ be corruption level and $\hat{\mathbf{y}}$ the corrupted (with outliers) version of $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, that is,

$$\hat{y}_i = \begin{cases} 100\ d_i & \text{with probability } r\\ y_i & \text{with probability } 1-r \end{cases} \tag{21}$$

where $\mathbf{d} \sim \mathcal{N}(0, \mathbf{I})$. We consider $r = 0$ and $r = 0.05$ (referred to as 5%). The number of samples for training, cross validation and testing are 40, 40 and 200, respectively. The results are shown in Table 1 (averaged over 50 repetitions) and in Figure 4. The outliers detected in one of the examples are shown in Figure 3.

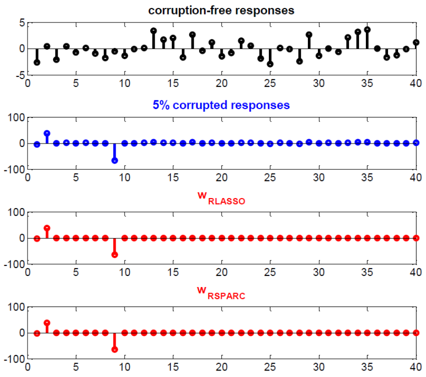


Fig. 3. Corruption-free and corrupted responses, and detected $\mathbf{w}_{\text{RLASSO}}$ and $\mathbf{w}_{\text{RSPARC}}$ obtained by RLASSO and RSPARC, respectively.

From Table 1 and Figure 2, we can easily see that when $r = 0$, the SPARC (RSPARC reduces to SPARC) performs the best; when $r = 5\%$, the non-robustified regularizers (LASSO, EN, OSCAR, SPARC) totally fail, while the RSPARC performs very well, in this problem much better than the RLASSO, although both of are able to correctly detect the corrupted responses (see Figure 3).

Finally, we compare the performances of RSPARC and RLASSO under different corruption levels. Let us define success when $\|\hat{\mathbf{x}} - \mathbf{x}\| \leq 10$, where $\hat{\mathbf{x}}$ is an estimate of $\mathbf{x}$, and keep the setup of above experiments unchanged. We run the RSPARC and RLASSO 50 times for each corruption level, obtaining the success rates and the mean DoF values plotted in Figure 4 (recall that in this problem, the DoF of $\mathbf{x}$ is 1); we can see that the RSPARC clearly outperforms RLASSO on this example.

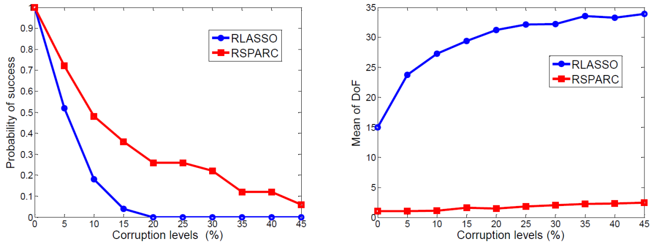


Fig. 4. Comparison on the performances of RSPARC and RLASSO under different corruption levels.

4. CONCLUSIONS

We have proposed a robust variant of our previously proposed *SPARsity and Clustering* regularization (RSPARC) for regression. We have shown that the proposed RSPARC is able to strongly and accurately promote group-sparsity and be robust to outliers. Future work will involve considering robust variants of SPARC for classification.

5. REFERENCES

[1] D. Lorenz and N. Worliczek, "Necessary conditions for variational regularization schemes," *Inverse Problems*, vol. 29, 2013.

	MAPE		MSPE		DoF		SER (%)	
Corruption levels	0%	5%	0%	5%	0%	5%	0%	5%
LASSO	33.5808	455.8603	9.2661	2441.9	14.94	29.78	0.85	72.05
EN	18.4127	417.9748	2.7645	1973.4	28.52	38.15	13.80	83.30
OSCAR	44.2015	437.7451	15.979	2285.2	4.94	33.20	1.40	78.05
SPARC	**12.1813**	427.5622	**1.3145**	2099.4	**1.70**	13.50	**0.60**	48.70
RLASSO	33.9713	37.2425	9.4919	11.418	15.08	15.36	0.82	1.30
RSPARC	12.3980	**16.8166**	1.3450	**2.4953**	1.84	**3.46**	**0.60**	**0.90**

Table 1. Results of the metrics obtained by different regularizations

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Structured sparsity through convex optimization," *Statistical Science*, vol. 27, no. 4, pp. 450–468, 2012.

[3] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society (B)*, vol. 68, pp. 49–67, 2005.

[4] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society (B)*, vol. 67, pp. 91–108, 2004.

[5] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society (B)*, vol. 67, pp. 301–320, 2005.

[6] H.D. Bondell and B.J. Reich, "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar," *Biometrics*, vol. 64, pp. 115–123, 2007.

[7] L.W. Zhong and J.T. Kwok, "Efficient sparse modeling with automatic feature grouping," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1436–1447, 2012.

[8] X. Zeng and M. A. T. Figueiredo, "Sparsity and clustering regularization for regression," in *Workshop on Signal processing with Adaptive Sparse Structured Representations*, 2013.

[9] X. Zeng and M. A. T. Figueiredo, "A novel sparsity and clustering regularization," in *19th Portuguese Conference on Pattern Recognition*, 2013.

[10] E. Candes and P. Randall, "Highly robust error correction byconvex programming," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 2829–2840, 2008.

[11] J. Laska, M. Davenport, and R. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice," in *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 1556–1560.

[12] K. Mitra, A. Veeraraghavan, and R. Chellappa, "Robust regression using sparse learning for high dimensional parameter estimation problems," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 3846–3849.

[13] Y. Jin and B. Rao, "Algorithms for robust linear regression by exploiting the connection to sparse signal recovery," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 3830–3833.

[14] C. Studer and R. Baraniuk, "Stable restoration and separation of approximately sparse signals," *arXiv preprint arXiv:1107.0420*, 2011.

[15] K. Mitra, A. Veeraraghavan, and R. Chellappa, "Analysis of sparse regularization based robust regression approaches," *IEEE Transactions on Signal Processing*, vol. 61, pp. 1249–1257, 2013.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, pp. 1–122, 2011.

[17] M.V. Afonso, J.M. Bioucas-Dias, and M.A.T. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, pp. 681–695, 2011.

[18] A. Kyrillidis and V. Cevher, "Combinatorial selection and least absolute shrinkage via the clash algorithm," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*. IEEE, 2012, pp. 2216–2220.

[19] X. Zeng and M. A. T. Figueiredo, "Solving OSCAR regularization problems by proximal splitting algorithms," *arXiv preprint arXiv:1309.6301*, 2013.

[20] X. Zeng and M. A. T. Figueiredo, "Atomic norm formulation of decreasing weighted sorted ℓ_1 regularization," *submitted*, 2014.

[21] P. Brucker, "An O(n) algorithm for quadratic knapsack problems," *Operations Research Letters*, vol. 3, no. 3, pp. 163–166, 1984.

[22] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 272–279.

[23] N. Maculan and G. Galdino de Paula, "A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$," *Operations Research Letters*, vol. 8, no. 4, pp. 219–222, 1989.

[24] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[25] J. Bioucas-Dias and M. A. T. Figueiredo, "A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, pp. 2992–3004, 2007.

[26] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2479–2493, 2009.