Adaptive LASSO based on joint M-estimation of regression and scale

Esa Ollila
Aalto University, Dept. of Signal Processing and Acoustics, P.O.Box 13000, FI-00076 Aalto, Finland

***Abstract*—The adaptive Lasso (Least Absolute Shrinkage and Selection Operator) obtains oracle variable selection property by using cleverly chosen adaptive weights for regression coefficients in the ℓ_1-penalty. In this paper, in the spirit of M-estimation of regression, we propose a class of adaptive M-Lasso estimates of regression and scale as solutions to generalized zero subgradient equations. The defining estimating equations depend on a differentiable convex loss function and choosing the LS-loss function yields the standard adaptive Lasso estimate and the associated scale statistic. An efficient algorithm, a generalization of the cyclic coordinate descent algorithm, is developed for computing the proposed M-Lasso estimates. We also propose adaptive M-Lasso estimate of regression with preliminary scale estimate that uses a highly-robust bounded loss function. A unique feature of the paper is that we consider complex-valued measurements and regression parameter. Consistent variable selection property of the adaptive M-Lasso estimates are illustrated with a simulation study.**

***Index Terms*—Adaptive Lasso, M-estimation, penalized regression, sparsity, variable selection**

I. Introduction

We consider the complex-valued linear model $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\mathbf{\Phi}$ is a known $n \times p$ complex-valued measurement matrix (or matrix of predictors), $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is the unknown vector of complex-valued regression coefficients (or system parameters) and $\boldsymbol{\varepsilon} \in \mathbb{C}^n$ denotes the additive noise. For ease of exposition, we consider the centered linear model (i.e., we assume that the intercept is equal to zero). The primary interest is to estimate the unknown parameter $\boldsymbol{\beta}$ given $\mathbf{y} \in \mathbb{C}^n$ and $\mathbf{\Phi} \in \mathbb{C}^{n \times p}$. When the linear system is *underdetermined* ($p > n$) or $p \approx n$, the least squares estimate (LSE) $\hat{\beta}_{\text{LS}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}\|_2^2$ does not have a unique solution or is subject to a very high variance. Furthermore, for large number of predictors, one wish to find a *sparse solution*, meaning that $\hat{\beta}_j = 0$ for most $j \in \{1, \ldots, p\}$, so that only the predictors that exhibit the strongest effects are selected. A common approach in the above cases it to use penalized/regularized regression with sparsity enforcing ℓ_1-penalty as in Lasso [1]. The Lasso, however, inherits the non-robustness (sensitivity to outliers) of LSE as well as its inefficiency when the noise follows a heavy-tailed non-Gaussian distribution.

The adaptive Lasso [2] uses adaptive weights for penalizing different coefficients in the ℓ_1-penalty. The weighted Lasso solves a weighted ℓ_1-penalized LS regression problem,

$$\underset{\boldsymbol{\beta} \in \mathbb{C}^p}{\text{minimize}} \Big\{ \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}\|_2^2 + \lambda \|\mathbf{w} \circ \boldsymbol{\beta}\|_1 \Big\} \qquad (1)$$

where $\lambda > 0$ is the shrinkage (penalty) parameter, chosen by the user, $\mathbf{w} = (w_1, \ldots, w_p)^\top$ is a vector of non-negative weights, and $\circ$ is the Hadamard product, i.e., the component-wise product of two vectors. Thus $\|\mathbf{w} \circ \boldsymbol{\beta}\|_1 = \sum_{j=1}^p w_j |\beta_j|$. Standard Lasso is obtained when $w_j \equiv 1$. Adaptive Lasso was proposed in the real-valued case, but it can be extended to complex-valued case in straightforward manner. Adaptive Lasso is obtained when $\lambda = \lambda_n$ depends on the sample size n and the weights are data dependent, defined as $\hat{w}_j = 1/|\hat{\beta}_{\text{init},j}|^\gamma$ for $\gamma > 0$, where $\hat{\boldsymbol{\beta}}_{\text{init}} \in \mathbb{C}^p$ is a root-n-consistent initial estimator to $\boldsymbol{\beta}$. It was shown in [2] that if $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$, then the adaptive Lasso estimate enjoys *oracle properties* (consistent variable selection and the same asymptotic normal distribution as the LSE that knows the true model). It should be noted that the root-n consistency of $\hat{\boldsymbol{\beta}}_{\text{init}}$ can be relaxed, see [2] for discussion. In this paper, we use $\gamma = 1$ and the standard ($w_j \equiv 1$) Lasso estimate $\hat{\boldsymbol{\beta}}_\lambda$ as $\hat{\boldsymbol{\beta}}_{\text{init}}$ as in [3].

The M-estimates of regression [4] are defined as solutions to generalized normal equations that depend on a score function which is the first derivative of the loss function $\rho(x)$, $\psi(x) = \rho'(x)$. Commonly used loss functions are the standard LS loss $\rho(x) = |x|^2$ or the robust Huber's loss function. Most robust loss and score functions require a preliminary estimate of the scale of the error distribution. In this paper, we propose a class of weighted/adaptive Lasso estimates following the spirit of M-estimation; namely, we define the weighted M-Lasso estimates of regression and scale as solutions to generalized zero subgradient equations that also depend on a score function. When the associated loss function is the LS-loss, these equations are a sufficient and necessary condition of a solution to the weighted Lasso problem (1). Furthermore, we develop a simple and efficient algorithm to compute the weighted M-Lasso estimate. This algorithm is a natural generalization of cyclic coordinate descent (CCD) algorithm [5] which is the current state-of-the-art method for computing the Lasso solution (1).

The paper is organized as follows. Robust loss functions and their properties are outlined in Section II. As examples we consider the Huber's loss and highly-robust (non-convex) Tukey's loss and introduce the notion of pseudo-residual vector. In Section III, we define the M-Lasso estimates of regression and scale and develop the generalized CCD algorithm for computing the solution. Section IV provides simulation studies to illustrate the model selection abilities and prediction accuracy of the proposed method in various noise conditions.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

Notations. The vector space $\mathbb{C}^n$ is equipped with the usual Hermitian inner product, $\langle \mathbf{a}, \mathbf{b}\rangle = \mathbf{a}^{\mathsf{H}}\mathbf{b}$, where $(\cdot)^{\mathsf{H}} = [(\cdot)^*]^\top$ denotes the Hermitian (complex conjugate) transpose. This induces the conventional (Hermitian) ℓ_2-norm $\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^{\mathsf{H}}\mathbf{a}}$. The ℓ_1-norm is the defined as $\|\mathbf{a}\|_1 = \sum_{i=1}^n |a_i|$, where $|a| = \sqrt{a^*a} = \sqrt{a_R^2 + a_I^2}$ denotes the modulus of a complex number $a = a_R + \jmath a_I$. For a matrix $\mathbf{A} \in \mathbb{C}^{n\times p}$, we denote by $\mathbf{a}_i \in \mathbb{C}^n$ its i^{th} column vector and $\mathbf{a}_{i\cdot} \in \mathbb{C}^p$ denotes the Hermitian transpose of its i^{th} row vector. Hence, we may write the measurement matrix $\mathbf{\Phi} \in \mathbb{C}^{n\times p}$ as $\mathbf{\Phi} = \begin{pmatrix}\boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_p\end{pmatrix} = \begin{pmatrix}\boldsymbol{\phi}_{1\cdot} & \cdots & \boldsymbol{\phi}_{n\cdot}\end{pmatrix}^{\mathsf{H}}$.

II. Robust loss functions and pseudo-residuals

Suppose that the noise terms ε_i are i.i.d. continuous random variables from a circular distribution [6] with p.d.f. $f(e) = (1/\sigma)f_0(e/\sigma)$, where $f_0(e)$ denotes the standard form of the density and $\sigma > 0$ is the scale parameter. If σ is known, then an M-estimator $\hat{\boldsymbol{\beta}}$ solves

$$-\sum_{i=1}^n \boldsymbol{\phi}_{i\cdot}\psi\left(\frac{y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\hat{\boldsymbol{\beta}}}{\sigma}\right) = \mathbf{0} \tag{2}$$

where $\psi : \mathbb{C} \to \mathbb{C}$, called the *score function*, is a complex conjugate derivative [7] of the loss function $\rho : \mathbb{C} \to \mathbb{R}_0^+$. As in [8], a function $\rho : \mathbb{C} \to \mathbb{R}_0^+$ is called a *loss function* if it is circularly symmetric (i.e., $\rho(e^{\jmath\theta}x) = \rho(x)\forall\,\theta \in \mathbb{R}$), $\mathbb{R}$-differentiable, increasing in $|x| > 0$ and satisfies $\rho(0) = 0$. Due to circularity assumption, $\rho(x) = \rho_0(|x|)$ for some $\rho_0 : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ and hence the score function becomes

$$\psi(x) = \frac{\partial}{\partial x^*}\rho(x) = \frac{1}{2}\left(\frac{\partial\rho}{\partial x_R} + \jmath\frac{\partial\rho}{\partial x_I}\right) = \frac{1}{2}\rho_0'(|x|)\mathrm{sign}(x),$$

where

$$\mathrm{sign}(x) = \begin{cases} x/|x|, & \text{for } x \neq 0 \\ 0, & \text{for } x = 0 \end{cases}$$

is the complex signum function and ρ_0' denotes the real derivative of the real-valued function ρ_0.

An objective function approach for M-estimation, on the other hand, defines an M-estimate of regression (again assuming σ is known) as a solution to an optimization program

$$\underset{\boldsymbol{\beta}\in\mathbb{C}^p}{\mathrm{minimize}} \sum_{i=1}^n \rho\left(\frac{y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\boldsymbol{\beta}}{\sigma}\right). \tag{3}$$

Naturally, if ρ is a convex loss function, then all solutions of (2) are solutions of (3). The maximum likelihood (ML-)estimate of regression is found by solving (3) with $\rho(x) = -\ln f_0(x)$ or equivalently, solving (2) with $\psi(x) = -\frac{\partial}{\partial x^*}\ln f_0(x)$.

In the complex-valued case, *Huber's [4] loss function* is defined as [8]:

$$\rho_{\mathrm{H},c}(x) = \begin{cases} |x|^2, & \text{for } |x| \leq c \\ 2c|x| - c^2, & \text{for } |x| > c, \end{cases} \tag{4}$$

where c is a user-defined *threshold* that influences the degree of robustness and efficiency of the method. Huber's loss function is a hybrid of ℓ_2 and ℓ_1 loss functions $\rho(x) = |x|^2$ and $\rho(x) = |x|$, respectively, using ℓ_2-loss for relatively small errors and ℓ_1-loss for relatively large errors. Moreover, it is convex. Huber's score function becomes

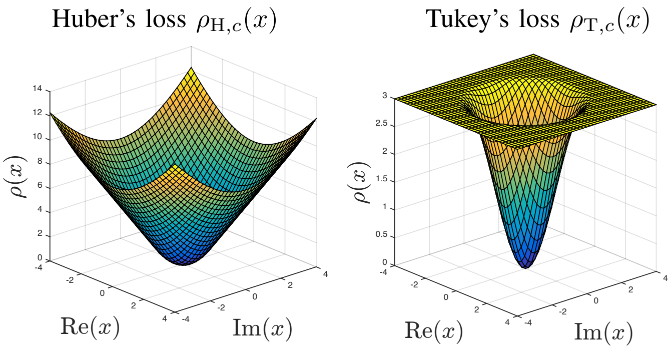

Fig. 1. Surface plots of the robust loss functions

$$\psi_{\mathrm{H},c}(x) = \begin{cases} x, & \text{for } |x| \leq c \\ c\,\mathrm{sign}(x), & \text{for } |x| > c \end{cases}.$$

We use $c = 1.215$ as our default choice which gives approximate 95% efficiency at the complex Gaussian noise.

Tukey biweight function is another commonly used loss function [4]. We define it for complex-values measurements as

$$\rho_{\mathrm{T},c}(x) = (c^2/3)\min\left\{1, 1 - \left(1 - (|x|/c)^2\right)^3\right\}.$$

Tukey's loss function is bounded, which makes it very robust to large outliers. As a consequence, it is also non-convex. The respective score function is

$$\psi_{\mathrm{T},c}(x) = \begin{cases} x\left(1 - (|x|/c)^2\right)^2 & \text{for } |x| \leq c \\ 0, & \text{for } |x| > c \end{cases}.$$

Thus large residuals $r_i = y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\boldsymbol{\beta}$ are completely rejected, i.e., they have zero weight in (2). For Tukey's loss function, we use $c = 3.0$ as our default choice which gives approximate 85% efficiency at the complex Gaussian noise. Huber's and Tukey's loss functions are depicted in Figure 1.

Let $\boldsymbol{r} \equiv \boldsymbol{r}(\boldsymbol{\beta}) = \mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}$ denote a residual vector for some candidate $\boldsymbol{\beta} \in \mathbb{C}^p$. The loss function then defines a *pseudo-residual*,

$$\boldsymbol{r}_\psi \equiv \boldsymbol{r}_\psi(\boldsymbol{\beta}, \sigma) = \psi\left(\frac{\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}}{\sigma}\right)\sigma, \tag{5}$$

where ψ-function in (5) acts coordinate-wise to vector $\boldsymbol{r}/\sigma$, so $[\psi(\boldsymbol{r}/\sigma)]_i = \psi(r_i/\sigma)$. Note that if $\rho(\cdot)$ is the conventional LS-loss, $\rho(x) = |x|^2$, then $\psi(x) = x$, and $\boldsymbol{r}_\psi$ is equal to the residual vector, $\boldsymbol{r}_\psi = \mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta} = \boldsymbol{r}$. The multiplier σ in (5) is essential in bringing the residuals back to the original scale of the data. Using the above notation, we may write (2) more compactly as $-\boldsymbol{\phi}_j^{\mathsf{H}}\boldsymbol{r}_\psi(\hat{\boldsymbol{\beta}}, \sigma) = 0$ for $j = 1, \ldots, p$, which will be elemental in our developments later on.

The discussion above assumes σ is known. In practise, σ is unknown and utilizing robust loss functions above requires a preliminary robust scale estimate $\hat{\sigma}$. Obtaining an accurate estimate of scale is a challenging problem in sparse regression scenario and therefore we develop a weighted/adaptive M-Lasso method in which regression and scale parameters are estimated jointly.

III. Weighted M-Lasso estimation

A. *Definition and properties*

Note that the LS criterion function $J_{\ell_2}(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2$ in (1) is convex (in fact strictly convex if $n > p$) and $\mathbb{R}$-differentiable but the ℓ_1-penalty function $\|\mathbf{w} \circ \boldsymbol{\beta}\|_1$ is not $\mathbb{R}$-differentiable at a point where at least one coordinate β_j is zero (and w_j is nonzero). However, we can resort to generalization of notion of gradient applicable for convex functions, called the *subdifferential* [9]. For a complex function $f : \mathbb{C}^p \to \mathbb{R}$ we can define subdifferential at a point $\boldsymbol{\beta}$ as

$$\partial f(\boldsymbol{\beta}) = \{\mathbf{z} \in \mathbb{C}^p : f(\boldsymbol{\beta}') \geq f(\boldsymbol{\beta}) + 2\mathrm{Re}(\langle \mathbf{z}, \boldsymbol{\beta}' - \boldsymbol{\beta}\rangle) \\ \text{for all } \boldsymbol{\beta}' \in \mathbb{C}^p\}.$$

Any element $\mathbf{z} \in \partial f(\boldsymbol{\beta})$ is then called a *subgradient* of f at $\boldsymbol{\beta}$. The subdifferential of the modulus $|\beta_j|$ is

$$\partial|\beta_j| = \begin{cases} \frac{1}{2}\mathrm{sign}(\beta_j), & \text{for } \beta_j \neq 0 \\ \frac{1}{2}s & \text{for } \beta_j = 0 \end{cases}$$

where s is some complex number verifying $|s| \leq 1$. Thus subdifferential of $|\beta_j|$ is the usual complex conjugate derivative when $\beta_j \neq 0$, i.e., $\partial|\beta_j| = \frac{\partial}{\partial \beta_j^*}|\beta_j|$ for $\beta_j \neq 0$. Then a necessary and sufficient condition for a solution to the Lasso problem (1) is that $\partial(J_{\ell_2}(\boldsymbol{\beta}) + \lambda\|\mathbf{w} \circ \boldsymbol{\beta}\|_1) \in \mathbf{0}$ which gives zero subgradient weighted Lasso equations

$$-\boldsymbol{\phi}_j^{\mathsf{H}}\big(\mathbf{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}\big) + \lambda w_j \hat{s}_j = 0 \quad \text{for } j = 1, \ldots, p \tag{6}$$

where $\hat{s}_j$ is 2 times an element of the subdifferential of $|\beta_j|$ evaluated at $\hat{\beta}_j$, i.e., equal to $\mathrm{sign}(\hat{\beta}_j)$ if $\hat{\beta}_j \neq 0$ and some complex number lying inside the unit complex circle otherwise. If $\hat{\boldsymbol{\beta}}_\lambda$ denotes a solution to weighted Lasso solution problem (1), then a natural estimate of the scale σ is

$$\hat{\sigma}_\lambda^2 = \frac{1}{n}\sum_{i=1}^{n} \big|y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\hat{\boldsymbol{\beta}}_\lambda\big|^2 = \frac{1}{n}\|\hat{\boldsymbol{r}}\|_2^2, \tag{7}$$

where $\hat{\boldsymbol{r}} = \mathbf{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}_\lambda$ denote the residual vector at the solution.

Definition 1. *Let* $\mathbf{w}$ *and* λ *denote a vector of non-negative weights and penalty parameter, respectively. Let* $\rho(x) = \rho_0(|x|)$ *denote a convex loss function. The* weighted M-Lasso estimates $(\hat{\boldsymbol{\beta}}_\lambda, \hat{\sigma}_\lambda) \in \mathbb{C}^p \times \mathbb{R}^+$ *are defined as solutions to generalized (zero subgradient) Lasso estimating equations,*

$$-\boldsymbol{\phi}_j^{\mathsf{H}}\boldsymbol{r}_\psi(\hat{\boldsymbol{\beta}}, \hat{\sigma}) + \lambda w_j \hat{s}_j = 0 \quad \textit{for } j = 1, \ldots, p \tag{8}$$

$$\alpha n - \sum_{i=1}^{n} \chi\left(\frac{|y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\hat{\boldsymbol{\beta}}|}{\hat{\sigma}}\right) = 0 \tag{9}$$

where $\alpha > 0$ *is a fixed scaling factor and the function* $\chi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ *is defined as*

$$\chi(t) = \rho_0'(t)t - \rho_0(t). \tag{10}$$

Equations (8) *and* (9) *are referred to as* weighted Lasso M-estimating equations.

Our approach to robust Lasso is different compared to earlier approaches in the literature which have followed the objective function approach to M-estimation by adding an ℓ_1-penalty to (3). We follow the more general estimating equation approach of M-estimation stated in (2).

Some remarks of this definition are in order. First, if one uses the LS-loss $\rho(x) = |x|^2$, then $\hat{\boldsymbol{r}}_\psi = \hat{\boldsymbol{r}}$ and (8) reduces to (6). Furthermore, since $\rho_0(t) = t^2$ and $\rho_0'(t) = 2t$, the χ-function in (10) is $\chi(t) = t^2$, and (9) reduces to (7) for $\alpha = 1$. In other words, for LS-loss, the weighted M-Lasso solution $(\hat{\boldsymbol{\beta}}_\lambda, \hat{\sigma}_\lambda)$ is the standard Lasso estimate $\hat{\boldsymbol{\beta}}_\lambda$ solving (1) and the standard scale statistic given by (7). Second, if $\lambda = 0$ (so no penalization) and $n > p$, then the solution to weighted M-Lasso estimating equations (8) and (9) is the unique solution to the convex optimization problem

$$\arg\min_{\boldsymbol{\beta},\sigma}\left\{Q(\boldsymbol{\beta}, \sigma) = \alpha n \sigma + \sum_{i=1}^{n} \rho\left(\frac{y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}}\boldsymbol{\beta}}{\sigma}\right)\sigma\right\}. \tag{11}$$

The objective function $Q(\boldsymbol{\beta}, \sigma)$ was proposed by Huber [4] in the real-valued case for joint M-estimation of regression and scale. Lasso penalized Huber's criterion was considered in [10] and ℓ_0-penalization in the complex-valued case in [8].

To simplify notation we write the pseudo-residual vector $\boldsymbol{r}_\psi(\hat{\boldsymbol{\beta}}, \hat{\sigma})$ in (8) as $\hat{\boldsymbol{r}}_\psi$. Then note that (8) can be written compactly as $\langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}}_\psi\rangle = \lambda w_j \hat{s}$ for $j = 1, \ldots, p$. Thus, after taking modulus of both sides, we obtain

$$|\langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}}_\psi\rangle| = \lambda w_j, \quad \text{if } \hat{\beta}_j \neq 0 \tag{12}$$

$$|\langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}}_\psi\rangle| \leq \lambda w_j, \quad \text{if } \hat{\beta}_j = 0. \tag{13}$$

In other words, whenever a component, say $\hat{\beta}_j$, of $\hat{\boldsymbol{\beta}}$ becomes non-zero, the corresponding absolute correlation between the pseudo-residual $\hat{\boldsymbol{r}}_\psi$ and the column $\boldsymbol{\phi}_j$ of $\boldsymbol{\Phi}$, $|\langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}}_\psi\rangle|$, meets the boundary λw_j in magnitude. This is a well-known property of Lasso; see e.g., [11] or [12] in the complex-valued case. This property is then fulfilled by weighted M-Lasso estimates by definition. In the real-valued case, [10] considered minimization of (non-weighted) penalized Huber's criterion $Q_\lambda(\boldsymbol{\beta}, \sigma) = Q(\boldsymbol{\beta}, \sigma) + \lambda\|\boldsymbol{\beta}\|_1$. The solution of $\min_{\boldsymbol{\beta},\sigma} Q_\lambda(\boldsymbol{\beta}, \sigma)$, however, is different from solutions to (8)-(9). This can be verified by noting that the zero subgradient equation $\partial_{\boldsymbol{\beta}} Q_\lambda(\boldsymbol{\beta}, \sigma) = \mathbf{0}$ is different from (8) for $w_j \equiv 1$. This also means that solution for weighted penalized Huber's criterion based on LS-loss function is not the weighted Lasso solution (1). This is somewhat counterintuitive. The equivalence with weighted Lasso solution to (1) and weighted M-Lasso for LS-loss, however, holds.

The scaling factor α in (9) is chosen so that the obtained scale estimate $\hat{\sigma}$ is Fisher-consistent for the unknown scale σ when $\{\varepsilon_i\}_{i=1}^{n} \overset{iid}{\sim} \mathbb{C}\mathcal{N}(0, \sigma^2)$. Due to (9), we choose it as $\alpha = \mathbb{E}[\chi(\varepsilon)]$, when $\varepsilon \sim \mathbb{C}\mathcal{N}(0, 1)$. For Huber's function (4) the χ-function in (10) becomes

$$\chi_{\mathrm{H},c}(|x|) = |\psi_{\mathrm{H},c}(x)|^2 = \begin{cases} |x|^2, & \text{for } |x| \le c \\ c^2, & \text{for } |x| > c \end{cases}. \tag{14}$$

In this case the estimating equation (9) can be written as

$$\sum_{i=1}^{n} \left| \psi_{\mathrm{H},c}\left(\frac{y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}} \hat{\boldsymbol{\beta}}}{\hat{\sigma}} \right) \hat{\sigma} \right|^2 = \hat{\sigma}^2 n\alpha \Leftrightarrow \hat{\sigma}^2 = \frac{1}{n\alpha} \left\| \hat{\boldsymbol{r}}_{\psi} \right\|_2^2,$$

where $\hat{\boldsymbol{r}}_{\psi} = \boldsymbol{r}_{\psi}(\hat{\boldsymbol{\beta}}, \hat{\sigma})$. The consistency factor $\alpha = \alpha(c)$ can be easily solved in closed-form by elementary calculus as $\alpha = c^2(1 - F_{\chi^2_2}(2c^2)) + F_{\chi^2_4}(2c^2)$, where $F_{\chi^2_k}$ denotes the c.d.f. of χ^2_k-distribution. Note that threshold parameter c determines the value of α.

B. Algorithm

Next we develop a simple and efficient algorithm, generalized cyclic coordinate descent (CCD) [5] algorithm, for computing the weighted M-Lasso estimates. Recall that CCD algorithm repeatedly cycles through the predictors updating one coordinate β_j at a time ($j = 1, \ldots, p$) while keeping others fixed at their current iterate values. At jth step, the update for $\hat{\beta}_j$ is obtained by soft-thresholding a conventional coordinate descent update $\hat{\beta}_j + \langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}} \rangle$, where $\hat{\boldsymbol{r}}$ denotes the residual vector $\hat{\boldsymbol{r}} = \boldsymbol{r}(\hat{\boldsymbol{\beta}})$ at current estimate $\hat{\boldsymbol{\beta}}$. In the complex-valued case, it is easy to verify (proof omitted) that

$$\mathrm{soft}_{\lambda}(y) = \arg\min_{\beta \in \mathbb{C}} \left\{ \frac{1}{2}|y - \beta|^2 + \lambda|\beta| \right\} = \mathrm{sign}(y)(|y| - \lambda)_+$$

is the complex soft-thresholding operator, where $(t)_+ = \max(t, 0)$. For weighted M-Lasso, similar updates are performed but $\hat{\boldsymbol{r}}$ is replaced by pseudo-residual vector $\hat{\boldsymbol{r}}_{\psi}$ and the update for scale is calculated prior to cycling through the coefficients.

The **Generalized CCD (GCCD) algorithm** for computing the weighted M-Lasso solutions proceeds as follows:

1) Update the scale $\hat{\sigma}^2 \leftarrow \dfrac{\hat{\sigma}^2}{\alpha n} \displaystyle\sum_{i=1}^{n} \chi\left(\frac{|y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}} \hat{\boldsymbol{\beta}}|}{\hat{\sigma}} \right)$
2) For $j = 1, \ldots, p$ do
    a) Update the pseudoresidual: $\hat{\boldsymbol{r}}_{\psi} \leftarrow \psi\left(\dfrac{\mathbf{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}}{\hat{\sigma}} \right) \hat{\sigma}$
    b) Update the coefficient: $\hat{\beta}_j \leftarrow \mathrm{soft}_{\lambda w_j}\left(\hat{\beta}_j + \langle \boldsymbol{\phi}_j, \hat{\boldsymbol{r}}_{\psi} \rangle \right)$
3) Repeat Steps 1 and 2 until convergence

We define adaptive M-Lasso estimates $(\hat{\boldsymbol{\beta}}_{\lambda}^{ad}, \hat{\sigma}_{\lambda}^{ad})$ simply as a weighted M-Lasso solution using data dependent weights $\hat{w}_j$-s and penalty λ_n as in [2]. They are computed using a two-stage procedure:

A1 Compute (non-weighted, so $w_j \equiv 1$) M-Lasso estimate $(\hat{\boldsymbol{\beta}}_{\lambda^*}, \hat{\sigma}_{\lambda^\star})$ where $\lambda^\star$ denotes the optimal penalty parameter chosen using the Bayesian information criterion (BIC) over a grid of λ values.

A2 Compute weights $\hat{w}_j = 1/|\hat{\beta}_{\mathrm{init},j}|$, where $\hat{\boldsymbol{\beta}}_{\mathrm{init}} = \hat{\boldsymbol{\beta}}_{\lambda^\star}$ and solve $(\hat{\boldsymbol{\beta}}_{\lambda}^{ad}, \hat{\sigma}_{\lambda}^{ad})$ as weighted M-Lasso solutions using $\hat{\mathbf{w}} = (\hat{w}_1, \ldots, \hat{w}_n)^{\top}$ and a penalty parameter λ_n.

In the simulation studies, we use $\lambda_n = \log(\log(n))$ which verifies condition on λ_n in [2, Th. 2]. Note that we can omit the variables for which $\hat{\beta}_{\mathrm{init},j} = 0$ and simply set $\hat{\beta}_{\lambda,j}^{ad} = 0$. BIC value is determined as $\lambda^\star = \arg\min_{\lambda \in [\lambda]} \{ 2n \ln \hat{\sigma}_{\lambda} + \mathsf{df}(\lambda) \cdot \ln n \}$ where $\mathsf{df}(\lambda)$ is the number of nonzero elements in $\hat{\boldsymbol{\beta}}_{\lambda}$ and $[\lambda]$ denotes a grid of λ values.

C. Adaptive M-Lasso with preliminary scale

If an accurate robust preliminary scale estimate $\hat{\sigma}_0$ is available, one may drop the assumption of convexity of the loss function in Definition 1 to allow for bounded (highly-robust) loss functions. Thus we define a weighted M-Lasso estimate $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{\lambda}$ with preliminary scale $\hat{\sigma}_0$ as a solution to

$$-\boldsymbol{\phi}_j^{\mathsf{H}} \boldsymbol{r}_{\psi}(\hat{\boldsymbol{\beta}}, \hat{\sigma}_0) + \lambda w_j \hat{s}_j = 0 \quad \text{for } j = 1, \ldots, p. \tag{15}$$

We then use Tukey's loss function $\rho_{\mathrm{T},c}(x)$ and compute the solution using the following three-stage procedure. First stage is as Step A1 for adaptive M-Lasso, where we utilise Huber's loss function in finding $(\hat{\boldsymbol{\beta}}_{\lambda^*}, \hat{\sigma}_{\lambda^\star})$. At second stage, one computes preliminary scale estimate as the *median absolute deviation (MAD)* of the residuals based on the Huber M-Lasso fit $\hat{\boldsymbol{\beta}}_{\lambda^\star}$ computed earlier, so

$$\hat{\sigma}_0 = 1.20112 \cdot \mathrm{median}\{ |y_i - \boldsymbol{\phi}_{i\cdot}^{\mathsf{H}} \hat{\boldsymbol{\beta}}_{\lambda^\star}| \}_{i=1}^{n}.$$

The scaling constant 1.20112 is used to obtain consistent scale estimate in complex Gaussian noise. At the last stage, we compute adaptive weights $\hat{w}_j = 1/|\hat{\beta}_{\mathrm{init},j}|$, where $\hat{\boldsymbol{\beta}}_{\mathrm{init}} = \hat{\boldsymbol{\beta}}_{\lambda^\star}$ and find the weighted M-Lasso solution with preliminary scale $\hat{\sigma}_0$ using Tukey's loss function, $\hat{w}_j$-s as weights and penalty λ_n. When computing the solution using the GCCD algorithm it is important to give $\hat{\boldsymbol{\beta}}_{\lambda^\star}$ as an initial warm start for the algorithm. Due to the good warm start, the algorithm appears to converge in practise despite of non-convexity of Tukey's loss function. Note that Step 1 (scale update) of GCCD algorithm is now omitted since the scale $\hat{\sigma}_0$ is not estimated.

IV. Simulations

We set $p = 2^3 = 8$ and $n = 2^7 = 128$. The coefficient vector $\boldsymbol{\beta}$ has $|\beta_1| = 1.0$, $|\beta_2| = 1.5$, $|\beta_3| = 2.0$ and $|\beta_j| = 0$ for $4 \le j \le p$, and $\mathrm{Arg}(\beta_j) \overset{iid}{\sim} Unif(0, 2\pi)$, $j = 1, \ldots, p$ for each MC trial. The measurement vector $\mathbf{y}$ is generated according to the linear model where $x_{ij} \overset{iid}{\sim} \mathbb{C}\mathcal{N}(0,1)$ and the error terms ε_i are i.i.d. from either the complex circular Gaussian distribution $\mathbb{C}\mathcal{N}(0, \sigma^2)$ or the circular Cauchy distribution $\mathbb{C}t_1(0, \sigma)$, i.e., circular complex t-distribution [6] with $\nu = 1$ degrees of freedom. In the former case, the scale parameter σ is the variance and in the latter case (as the variance does not exist) the population MAD, $\sigma = \mathrm{Med}_F(|\varepsilon_i|)$. The *support* of $\boldsymbol{\beta}$ is the index set of its non-zero elements, i.e., $\Gamma = \mathrm{supp}(\boldsymbol{\beta}) = \{ j \in \{1, \ldots, p\} : \beta_j \neq 0 \}$ and $\hat{\Gamma} = \mathrm{supp}(\hat{\boldsymbol{\beta}})$ denotes the support of $\hat{\boldsymbol{\beta}}$. We consider the cases $\sigma = 0.5$ and $\sigma = 2$ which yield signal-to-noise ratio, $\mathrm{SNR} = 20 \log_{10}(\mathrm{ave}_{j \in \Gamma} |\beta_j| / \sigma) = 12$ dB and $\mathrm{SNR} = 0$ dB, respectively.

To assess the model selection abilities of the estimators, we calculate the *correct model selection* rate, $\mathrm{CMS}(\hat{\boldsymbol{\beta}}) = \mathsf{I}(\Gamma = \hat{\Gamma})$ and the *overfitting* rate, $\mathrm{OF}(\hat{\boldsymbol{\beta}}) = \mathsf{I}(\Gamma \subset \hat{\Gamma})$, where $\mathsf{I}(\cdot)$ denotes the indicator function. In the latter case, all the non-zero coefficients *and* at least one zero coefficient is selected. The underfitting rate, $\mathrm{UF}(\hat{\boldsymbol{\beta}}) = \neg(\mathrm{CMS}(\hat{\boldsymbol{\beta}}) \vee \mathrm{OF}(\hat{\boldsymbol{\beta}}))$, means that at least one significant predictor is excluded from the model. This is the least wanted scenario since adaptive Lasso can not improve upon the CMS rate of an underfitting initial estimate $\hat{\boldsymbol{\beta}}_{\text{init}}$. To measure the degree of overfit/underfit, we compute the (conditional) number of false positives/negatives, $\mathrm{FP}(\hat{\boldsymbol{\beta}}) = \#(\hat{\beta}_j \neq 0 \wedge \beta_j = 0)$ conditioned that $\mathrm{OF}(\hat{\beta}) = 1$ and $\mathrm{FN}(\hat{\boldsymbol{\beta}}) = \#(\hat{\beta}_j = 0 \wedge \beta_j \neq 0)$ conditioned that $\mathrm{UF}(\hat{\beta}) = 1$. The prediction accuracy is measured via *prediction error*,

$$\mathrm{PE}(\hat{\boldsymbol{\beta}}) = \begin{cases} \left(\frac{1}{n}\sum_{i=1}^{n} \left|\tilde{y}_i - \tilde{\boldsymbol{\phi}}_{i\cdot}^{\mathsf{H}}\hat{\boldsymbol{\beta}}\right|^2\right)^{1/2} & \varepsilon_i \sim \mathbb{C}\mathcal{N}(0,\sigma^2) \\ \mathrm{median}\left\{\left|\tilde{y}_i - \tilde{\boldsymbol{\phi}}_{i\cdot}^{\mathsf{H}}\hat{\boldsymbol{\beta}}\right|\right\}_{i=1}^{n} & \varepsilon_i \sim \mathbb{C}t_1(0,\sigma) \end{cases}$$

where an additional test data set $(\tilde{\mathbf{y}}, \tilde{\boldsymbol{\Phi}})$ of same sample size n is generated from the respective sampling schemes for each MC trial. Note that *median absolute prediction error* (MeAPE) is used for Cauchy noise since Cauchy distribution does not have finite variance. The above performance measures for the *oracle estimator*, which uses the true coefficient vector $\boldsymbol{\beta}$, is also given as a point of reference for the evaluated methods. All measures are computed as averages over 1000 MC trials.

Methods included in our study are, **Las**: standard ($w_j \equiv 1$) Lasso using BIC for penalty parameter selection, **Hub**: standard ($w_j \equiv 1$) M-Lasso estimate of regression and scale using Huber's loss and BIC, **adLas**: adaptive Lasso using **Las** as initial estimate $\hat{\boldsymbol{\beta}}_{\text{init}}$, **Hub**: adaptive M-Lasso estimate of of regression and scale using **Hub** as initial estimate $\hat{\boldsymbol{\beta}}_{\text{init}}$, **adTuk**: adaptive M-Lasso estimate with preliminary scale $\hat{\sigma}_0$ (that uses Tukey's loss function and **Hub** as $\hat{\boldsymbol{\beta}}_{\text{init}}$ as explained in Section III-C).

Table 1 reports the model selection performance measures for Gaussian (upper table) and Cauchy (lower table) noise. In both medium SNR ($\sigma = 0.5$) and low SNR ($\sigma = 2$) Gaussian cases, **Las** and **Hub** exhibit very similar performance. This is even more apparent when inspecting the prediction errors tabulated in Table II. In other words, using robust M-Lasso with Huber's loss in Gaussian noise leads to marginal loss in performance. In medium SNR Gaussian noise, all adaptive methods have oracle performance (full 100% CMS rates). In low SNR Gaussian noise, however, only **Tuk** maintains full 100% CMS rate, whereas CMS rates of **Las** and **Hub** drop down slightly to 98% and 99%, respectively. This illustrates that **Tuk** estimator based on bounded loss function can be useful even in Gaussian noise with low SNR. In heavy-tailed Cauchy noise, the performance of the non-robust **Las** and **adLas** completely collapse as expected. For example, in low SNR Cauchy noise, **Las** does underfitting in 99% of MC trials. Furthermore, FN number reveals that all zeros ($\hat{\boldsymbol{\beta}} = \mathbf{0}$) is the most commonly selected model. The robust M-Lasso methods, however, maintain excellent performance. In medium SNR Cauchy errors, both **adHub** and **adTuk** achieve oracle performance (full CMS rate) and in low SNR Cauchy noise, CMS rates decrease to 84% and 97%, respectively. In terms of PE-s of Table 2, **Hub** is seen to obtain the best performance both in Gaussian and Cauchy noise. Adaptive M-Lasso (using $\lambda_n = \log(\log(n)))$ does not improve on PE of the initial estimator, although it provides significant improvements in model selection performance.

	$\sigma = 0.5$				$\sigma = 2.0$			
Method	CM	OF	FP	FN	CM	OF	FP	FN
Oracle	100	0	0	0	100	0		0
Las	70	30	1.31	0	71	29	1.31	0
Hub	68	32	1.46	0	67	33	1.47	0
adLas	100	0	0	0	98	2	1.06	0
adHub	100	0	0	0	99	1	1.08	0
adTuk	100	0	0	0	100	0	0	0
Oracle	100	0	0	0	100	0	0	0
Las	33	12	1.34	2.61	1	0	1.67	2.94
Hub	29	71	1.92	0	31	68	1.81	1.00
adLas	37	8	1.17	2.61	1	0	1.67	2.94
adHub	100	0	0	0	84	15	1.15	1.00
adTuk	100	0	0	0	97	2	1.06	1.00

TABLE I
MODEL SELECTION PERFORMANCE OF DIFFERENT METHODS IN GAUSSIAN (UPPER TABLE) AND CAUCHY (LOWER TABLE) NOISE.

σ	Oracle	Las	Hub	adLas	adHub	adTuk
0.5	0.500	0.517	0.517	0.529	0.537	0.587
2.0	2.001	2.064	2.065	2.034	2.038	2.051
0.5	0.505	1.708	0.541	1.625	0.571	0.633
2.0	2.021	3.424	2.156	3.456	2.109	2.109

TABLE II
PREDICTION ERROR (PE) OF DIFFERENT METHODS IN GAUSSIAN (UPPER TABLE) AND CAUCHY (LOWER TABLE) NOISE.

REFERENCES

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc., Ser. B*, vol. 58, pp. 267–288, 1996.

[2] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[3] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[4] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[5] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Stat.*, vol. 1, no. 2, pp. 302–332, 2007.

[6] E. Ollila, J. Eriksson, and V. Koivunen, "Complex elliptically symmetric random variables – generation, characterization, and circularity tests," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 58–69, 2011.

[7] J. Eriksson, E. Ollila, and V. Koivunen, "Essential statistics and tools for complex random variables," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5400–5408, 2010.

[8] E. Ollila, "Multichannel sparse recovery of complex-valued signals using Huber's criterion," in *Proc. Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa'15)*, Pisa, Italy, Jun. 16 – 19, 2015, pp. 32–36.

[9] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[10] A. B. Owen, "A robust hybrid of lasso and ridge regression," *Contemporary Mathematics*, vol. 443, pp. 59–72, 2007.

[11] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

[12] P. Gerstoft, A. Xenaki, and C. Mecklenbrauker, "Multiple and single snapshot compressive beamforming," *J. Acoust. Soc. Am.*, vol. 138, no. 4, pp. 2003–2014, 2015.