ROBUST ITERATIVE HARD THRESHOLDING FOR COMPRESSED SENSING

Esa Ollila, Hyon-Jung Kim and Visa Koivunen

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

ABSTRACT

Compressed sensing (CS) or sparse signal reconstruction (SSR) is a signal processing technique that exploits the fact that acquired data can have a sparse representation in some basis. One popular technique to reconstruct or approximate the unknown sparse signal is the iterative hard thresholding (IHT) which however performs very poorly under non-Gaussian noise conditions or in the face of outliers (gross errors). In this paper, we propose a robust IHT method based on ideas from M-estimation that estimates the sparse signal and the scale of the error distribution simultaneously. The method has a negligible performance loss compared to IHT under Gaussian noise, but superior performance under heavy-tailed non-Gaussian noise conditions.

Index Terms— Compressed sensing, iterative hard thresholding, M-estimation, robust estimation

1. INTRODUCTION

The compressed sensing (CS) problem can be formulated as follows [1]. Let $\mathbf{y} = (y_1, \ldots, y_M)^\top$ denote the observed data (*measurements*) modelled as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{\Phi} = \begin{pmatrix} \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_M \end{pmatrix}^\top$ is $M \times N$ *measurement matrix* with more column vectors than row vectors $\boldsymbol{\phi}_i$ (i.e., $N > M$), $\mathbf{x} = (x_1, \ldots, x_N)^\top$ is the unobserved *signal vector* and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_M)^\top$ is the (unobserved) *random noise* vector. It is assumed that the signal vector $\mathbf{x}$ is K-*sparse* (i.e., it has K *non-zero* elements) or is *compressible* (i.e., it has a representation whose entries decay rapidly when sorted in a decreasing order) The signal *support* (i.e., the locations of non-zero elements) is denoted as $\Gamma = \text{supp}(\mathbf{x}) = \{j : x_j \neq 0\}$. Then, we aim to reconstruct or approximate the signal vector $\mathbf{x}$ by K-sparse representation knowing only the acquired vector $\mathbf{y}$, the measurement matrix $\mathbf{\Phi}$ and the sparsity K.

A K-sparse estimate of $\mathbf{x}$ can be found by solving the optimization $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2$ subject to $\|\mathbf{x}\|_0 \leq K$, where $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm, $\|\mathbf{x}\|_0 = \#\{j : x_j \neq 0\}$. This optimization problem is known to be NP-hard and hence

suboptimal approaches have been under active research; see [1] for a review. The widely used methods developed for estimating $\mathbf{x}$ such as *Iterative Hard Thresholding (IHT)* [2, 3] are shown to perform very well provided that suitable conditions (e.g., restricted isometry property on $\mathbf{\Phi}$ and non impulsive noise conditions) are met. Since the recovery bounds of IHT depend linearly on $\|\boldsymbol{\varepsilon}\|_2$, the method often fails to provide accurate reconstruction/approximation under the heavy-tailed or spiky non-Gaussian noise.

Despite the vast interest in CS/SSR during the past decade, *sparse and robust signal reconstruction methods* that are resistant to heavy-tailed non-Gaussian noises or outliers have appeared in the literature only recently; e.g, [4, 5, 6]. In [6] we proposed a robust IHT method using a robust loss function with a preliminary estimate of the scale, called the generalized IHT. The Lorentizian IHT (LIHT) proposed in [4] is a special case of our method using the Cauchy loss function. A major disadvantage of these methods is that they require a preliminary (auxiliary) robust estimate of the scale parameter σ of the error distribution. In this paper, we propose a novel IHT method that estimates $\mathbf{x}$ and σ simultaneously.

The paper is organized as follows. Section 2 provides a review of the robust M-estimation approach to regression using different robust loss/objective functions. We apply these approaches to obtain (constrained) sparse and robust estimates of $\mathbf{x}$ in the CS system model using the IHT technique. Section 3 describes the new robust IHT method and Section 4 provides extensive simulation studies illustrating the effectiveness of the method in reconstructing a K-sparse signal in various noise conditions and SNR regimes.

Notations: For a vector $\mathbf{a} \in \mathbb{R}^m$, a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ and an index set $\Gamma = (\gamma_1, \ldots, \gamma_p)$ with $p < m$, a_i denotes the ith component of $\mathbf{a}$, $\mathbf{a}_i$ denotes the ith column vector of $\mathbf{A}$ and $\mathbf{a}_\Gamma$ denotes the p-vector of $\mathbf{a}$ with elements a_{γ_i} selected according to the support set Γ. Similarly $\mathbf{A}_\Gamma = (\mathbf{a}_{\gamma_1} \cdots \mathbf{a}_{\gamma_p})$ is an $n \times p$ matrix whose columns are selected from the columns of $\mathbf{A}$ according to the index set Γ.

2. ROBUST REGRESSION AND LOSS FUNCTIONS

We assume that the noise terms ε_i are independent and identically distributed (i.i.d.) random variables from a continuous

symmetric distribution and let $\sigma > 0$ be the *scale parameter* of the error distribution. The density of ε_i is $f_\varepsilon(e) = (1/\sigma) f_0(e/\sigma)$, where $f_0(\cdot)$ is the standard form of the density, e.g., $f_0(e) = (1/\sqrt{2\pi}) \exp(-\frac{1}{2}e^2)$ in case of normal (Gaussian) error distribution. Let *residuals* for a given (candidate) signal vector $\mathbf{x}$ be $e_i \equiv e_i(\mathbf{x}) = y_i - \boldsymbol{\phi}_i^\top \mathbf{x}$ and write $\mathbf{e} \equiv \mathbf{e}(\mathbf{x}) = (e_1, \ldots, e_M)^\top = \mathbf{y} - \boldsymbol{\Phi}\mathbf{x}$ for the vector residuals. When $M > N$ and no sparse approximation is assumed for $\mathbf{x}$ (i.e., unconstrained overdetermined problem), (1) is just a conventional *regression model*. We start with a brief review of the robust M-estimation method.

A common approach to obtain a robust estimator of regression parameters is to replace the least squares (LS) or ℓ_2-loss function $\rho(e) = \frac{1}{2}e^2$ by a *robust loss function* which downweights large residuals. Suppose that we have obtained an (preliminary, a priori) estimate $\hat{\sigma}$ of the scale parameter σ. Then, a robust M-estimator $\hat{\mathbf{x}}$ of $\mathbf{x}$ can be obtained by solving the optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \rho\left(\frac{y_i - \boldsymbol{\phi}_i^\top \mathbf{x}}{\hat{\sigma}} \right) \tag{2}$$

where ρ is a continuous, even function increasing in $e \geq 0$. The M-*estimating equation* approach is to find $\hat{\mathbf{x}}$ that solves $\sum_{i=1}^{M} \psi\big((y_i - \boldsymbol{\phi}_i^\top \mathbf{x})/\hat{\sigma}\big)\boldsymbol{\phi}_i = \mathbf{0}$, where ψ is a continuous and odd function ($\psi(-e) = -\psi(e)$), referred to as a *score function*. When $\psi = \rho'$, a stationary point of the objective function in (2) is a solution to the estimating equation.

A commonly used loss function is **Huber's loss function** which combines ℓ_2 and ℓ_1 loss functions and is defined as

$$\rho_{\mathrm{H}}(e) = \begin{cases} \frac{1}{2}e^2, & \text{for } |e| \leq c \\ c|e| - \frac{1}{2}c^2, & \text{for } |e| > c, \end{cases} \tag{3}$$

where c is a user-defined *tuning constant* that influences the degree of robustness and efficiency of the method. The following choices, $c_1 = 1.345$ and $c_2 = 0.732$, yield 95 and 85 percent (asymptotic) relative efficiency compared to LSE of regression in case of Gaussian errors. Huber's loss function is differentiable and convex function, and the score function $\psi = \rho'$ is a *winsorizing (clipping, trimming) function*

$$\psi_{\mathrm{H}}(e) = \max[-c, \min(c, e)] = \begin{cases} e, & \text{for } |e| \leq c \\ c\,\mathrm{sign}(e), & \text{for } e > c \end{cases}$$

The smaller the c, the more downweighting (clipping) is done to the residuals.

3. ROBUST IHT

The problem of generalized IHT of [6] is in how to obtain an accurate and robust preliminary scale estimate $\hat{\sigma}$. To circumvent the above problem, we propose to estimate the $\mathbf{x}$ and σ

simultaneously (jointly). To do this elegantly, we propose to minimize

$$Q(\mathbf{x}, \sigma) = \sigma \sum_{i=1}^{M} \rho\left(\frac{y_i - \boldsymbol{\phi}_i^\top \mathbf{x}}{\sigma} \right) + (M - K)\alpha\sigma \tag{4}$$

subject to $\quad \|\mathbf{x}\|_0 \leq K,$

where ρ is a convex loss function which should verify $\lim_{|x|\to\infty} \rho(x)/|x| = c \leq \infty$ and $\alpha > 0$ is a scaling factor chosen so that the solution $\hat{\sigma}$ is Fisher-consistent for σ when $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. This is achieved by setting $\alpha = \mathbb{E}[\chi(u)]$, where $u \sim \mathcal{N}(0, 1)$ and $\chi(e) = \psi(e)e - \rho(e)$. Note that a multiplier $(M - K)$ is used in the second term of (4) instead of M in order to reduce the bias of the obtained scale estimate $\hat{\sigma}$ at small sample lengths. The objective function Q in (4) was proposed for joint estimation of regression and scale by Huber (1973) [7] and is often referred to as *"Huber's proposal 2"*. Note that $Q(\mathbf{x}, \sigma)$ is a convex function of $(\mathbf{x}, \sigma)$ which allows to derive a simple convergence proof of an iterative algorithm to compute the solution $(\hat{\mathbf{x}}, \hat{\sigma})$.

Let us choose Huber's loss function $\rho_{\mathrm{H}}(e)$ in eq. (3) as our choice of ρ function. In this case χ-function becomes $\chi_{\mathrm{H}}(e) = \frac{1}{2}\psi_{\mathrm{H}}^2(e)$ and the scaling factor $\alpha = \beta/2$ can be computed as

$$\beta = 2\{c^2(1 - F_G(c)) + F_G(c) - 1/2 - c\,f_G(c)\}, \tag{5}$$

where F_G and f_G denote the c.d.f and the p.d.f. of $\mathcal{N}(0, 1)$ distribution, respectively, and c is the downweighting threshold of Huber's loss function. The algorithm for finding the solution to (4) is given in Algorithm 1. Therein $H_K(\cdot)$ (in Step 5 and initialization step) denotes the hard thresholding operator that sets all but the largest (in magnitude) K elements of its vector-valued argument to zero.

Computing the stepsize μ^n in Step 4. Assuming we have identified the correct signal support at nth iteration, an optimal step size can be found in gradient ascent direction $\mathbf{x}_{\Gamma^n} + \mu^n \mathbf{g}_{\Gamma^n}$ by solving

$$\mu_{opt}^n = \arg\min_{\mu} \sum_{i=1}^{M} \rho\left(\frac{y_i - [\boldsymbol{\phi}_i]_{\Gamma^n}^\top (\mathbf{x}_{\Gamma^n} + \mu\mathbf{g}_{\Gamma^n})}{\sigma^{n+1}} \right). \tag{6}$$

Since closed-form solution can not be found for μ_{opt}^n, we aim at finding a good approximation in closed-form. By writing $v(e) = \rho(e)/e^2$, we can express the problem as

$$\mu_{opt}^n = \arg\min_{\mu} \sum_{i=1}^{M} v_i^n(\mu)\big(y_i - [\boldsymbol{\phi}_i]_{\Gamma^n}^\top (\mathbf{x}_{\Gamma^n} + \mu\mathbf{g}_{\Gamma^n})\big)^2 \tag{7}$$

where $v_i^n(\mu) = v\big((y_i - [\boldsymbol{\phi}_i]_{\Gamma^n}^\top (\mathbf{x}_{\Gamma^n} + \mu\mathbf{g}_{\Gamma^n}))/\sigma^{n+1}\big)$ depend on μ. If we replace $v_i^n(\mu)$ by its approximation $v_i^n = v_i^n(0)$, we can find stepsize (i.e., an approximation of μ_{opt}^n) in closed-form. Hence, when the iteration starts at $n = 0$, we calculate the stepsize μ^0 in Step 4 as

$$\mu^0 = \frac{(\mathbf{e}^0)^\top \mathbf{V}^0 \boldsymbol{\Phi}_{\Gamma^0} \mathbf{g}_{\Gamma^0}}{\mathbf{g}_{\Gamma^0}^\top \boldsymbol{\Phi}_{\Gamma^0}^\top \mathbf{V}^0 \boldsymbol{\Phi}_{\Gamma^0} \mathbf{g}_{\Gamma^0}}, \tag{8}$$

Algorithm 1: Huber IHT (HIHT) algorithm

Input: $\mathbf{y}$, $\boldsymbol{\Phi}$, sparsity K and trimming threshold c.
Output: $(\mathbf{x}^{n+1}, \sigma^{n+1}, \Gamma^{n+1})$ estimates of $\mathbf{x}$, σ and Γ.
Initialization: Set $\mathbf{x}^0 = \mathbf{0}$, $\sigma^0 = 1$. Compute the scaling factor $\beta = \beta(c)$ and the initial signal support $\Gamma^0 = \mathrm{supp}\big(H_K(\boldsymbol{\Phi}^\top \mathbf{y}_\psi)\big)$, where $\mathbf{y}_\psi = \psi_\mathrm{H}(\mathbf{y})$.

For $n = 0, 1, \ldots$, iterate the steps

1. *Compute the residuals* $\mathbf{e}^n = \mathbf{y} - \boldsymbol{\Phi}\mathbf{x}^n$

2. *Update the value of the scale*:
$$(\sigma^{n+1})^2 = \frac{(\sigma^n)^2}{(M-K)\beta} \sum_{i=1}^{M} \psi_\mathrm{H}^2\left(\frac{e_i^n}{\sigma^n}\right)$$

3. *Compute the pseudo-residual and the gradient update*
$$\mathbf{e}_\psi^n = \psi_\mathrm{H}\left(\frac{\mathbf{e}^n}{\sigma^{n+1}}\right)\sigma^{n+1} \text{ and } \mathbf{g} = \boldsymbol{\Phi}^\top \mathbf{e}_\psi^n$$

4. *Compute the stepsize* μ^n *using* (8) *if* $n = 0$ *and* (10) *otherwise*

5. *Update the value of the signal vector and the support*
$$\mathbf{x}^{n+1} = H_K(\mathbf{x}^n + \mu^n \mathbf{g}) \text{ and } \Gamma^{n+1} = \mathrm{supp}(\mathbf{x}^{n+1})$$

6. *Approve the updates* $(\mathbf{x}^{n+1}, \Gamma^n)$ *or recompute them* (discussed later).

Until $\frac{\|\mathbf{x}^{n+1} - \mathbf{x}^n\|^2}{\|\mathbf{x}^n\|^2} < \delta$, where δ is a predetermined tolerance/accuracy level (e.g., $\delta = 1.0^{-6}$).

---

where $\mathbf{V}^0 = \mathrm{diag}(v_1^0, \ldots, v_M^0)$. When iteration proceeds (for $n = 1, 2, \ldots$), the current support Γ^n and the signal update $\mathbf{x}^n$ are more accurate estimates of Γ and $\mathbf{x}$. Hence, when $n \geq 1$, we find an approximation of μ_{opt}^n by solving

$$\mu^n = \arg\min_\mu \sum_{i=1}^n w_i^n \left(y_i - [\boldsymbol{\phi}_i]_{\Gamma^n}^\top(\mathbf{x}_{\Gamma^n}^n + \mu \mathbf{g}_{\Gamma^n})\right)^2 \quad (9)$$

where the "weights" w_i^n are defined as $w_i^n = w_\mathrm{H}\big((y_i - [\boldsymbol{\phi}_i]_{\Gamma^n}^\top \mathbf{x}_{\Gamma^n}^n)/\sigma^{n+1}\big)$ with $w_\mathrm{H}(e) = \psi_\mathrm{H}(e)/e$ being the Huber's weight function. The solution to (9) is

$$\mu^n = \frac{\mathbf{g}_{\Gamma^n}^\top \mathbf{g}_{\Gamma^n}}{\mathbf{g}_{\Gamma^n}^\top \boldsymbol{\Phi}_{\Gamma^n}^\top \mathbf{W}^n \boldsymbol{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}}, \quad (10)$$

where $\mathbf{W}^n = \mathrm{diag}(w_1^n, \ldots, w_M^n)$.

Approving or recomputing the updates $(\mathbf{x}^{n+1}, \Gamma^{n+1})$ **in Step 6.** We accept the updates if $Q(\mathbf{x}^{n+1}, \sigma^{n+1}) < Q(\mathbf{x}^n, \sigma^n)$, otherwise we set $\mu^n \leftarrow \mu^n/2$ and go back to Step 5 and recompute new updates.

Relation to IHT algorithm. Consider the case that trimming threshold c is arbitrarily large ($c \to \infty$). Then it is easy to show that the proposed Huber IHT method coincides with IHT [2, 3]. This follows as Step 2 can be discarded as it does not have any effect on Step 3 because $\mathbf{e}_\psi^n = \mathbf{e}^n$ for very large c (as $\psi_\mathrm{H}(e) = e$). Furthermore, now $\mathbf{V}^0 = \mathbf{I}$ and $\mathbf{W}^n = \mathbf{I}$, so the optimal stepsizes (8) and (10) reduce to the one used in the normalized IHT algorithm [3].

4. SIMULATION STUDIES

Description of the setup and performance measures. The elements of the measurement matrix $\boldsymbol{\Phi}$ are drawn from $\mathcal{N}(0, 1)$ distribution after which the columns of are normalized to have unit norm. The K nonzero coefficients of $\mathbf{x}$ are set to have equal amplitude $\sigma_s = |x_i| = 10$ for all $i \in \Gamma$, equiprobable signs and $\Gamma = \mathrm{supp}(\mathbf{x})$ is randomly chosen from $\{1, \ldots, N\}$ without replacement for each trial. The *signal to noise ratio (SNR)* is $\mathrm{SNR}(\sigma) = 20\log_{10}(\sigma_s/\sigma)$ and depends on the scale parameter σ of the error distribution. In case of Gaussian errors, the scale equals the *standard deviation* (SD) $\sigma_\mathrm{SD} = \sqrt{\mathbb{E}[|\varepsilon|^2]}$, in case of Laplacian errors, the *mean absolute deviation* (MeAD) $\sigma_\mathrm{MeAD} = \mathbb{E}[|\varepsilon|]$, and in case of Student's t_ν-distribution with degrees of freedom (d.o.f.) $\nu \geq 0$, the *median absolute deviation* (MAD) $\sigma_\mathrm{MAD} = \mathrm{Med}(|\varepsilon|)$. Note that SD does not exist for t_ν-distribution with $\nu \leq 2$. As performance measures we use the *mean squared error* $\mathrm{MSE}(\hat{\mathbf{x}}) = \frac{1}{Q}\sum_{q=1}^Q \|\hat{\mathbf{x}}^{[q]} - \mathbf{x}^{[q]}\|_2^2$ and the *probability of exact recovery*

$$\mathrm{PER}(\hat{\mathbf{x}}) = \frac{1}{Q}\sum_{q=1}^Q \mathrm{I}(\hat{\Gamma}^{[q]} = \Gamma^{[q]}),$$

where $\mathrm{I}(\cdot)$ denotes the indicator function, $\hat{\mathbf{x}}^{[q]}$ and $\hat{\Gamma}^{[q]} = \mathrm{supp}(\hat{\mathbf{x}}^{[q]})$ denote the estimate of the K-sparse signal $\mathbf{x}^{[q]}$ and the signal support $\Gamma^{[q]}$ for the qth trial, respectively. The number of Monte-Carlo trials is $Q = 2000$, $M = 512$, $N = 256$ and the sparsity level is $K = 8$. The methods included in the study are **IHT** (referring to the normalized IHT method [3]), **LIHT** (referring to LIHT method of [4]) and **HIHT-c_i**, $i \in \{1, 2\}$ (referring to the Huber IHT method of Algorithm 1 using trimming thresholds $c_1 = 1.345$ and $c_2 = 0.732$).

Experiment I: Gaussian and Laplacian noise: Figure 1 depict the MSE as a function of $\mathrm{SNR}(\sigma)$ in the Gaussian $\mathcal{N}(0, \sigma_\mathrm{SD}^2)$ and Laplace $Lap(0, \sigma_\mathrm{MeAD})$ noise distribution cases, respectively. In the Gaussian case, the IHT has the best performance but HIHT-c_1 suffers only a negligible 0.2 dB performance loss. LIHT experienced convergence problems in the Gaussian errors simulation setup and hence was left out from this study. These problems may be due to the choice of the preliminary scale estimate $\hat{\sigma}$ used in LIHT which seems appropriate only for heavy-tailed distributions at high SNR regimes. In the case of Laplacian errors, the HIHT-c_2 has the best performance, next comes HIHT-c_1, whereas LIHT has

Table 1: PER rates of the methods under $t_\nu(0, \sigma_{\text{MAD}})$ distributed noise at different $\text{SNR}(\sigma_{\text{MAD}})$ and d.o.f. ν.

Method	Degrees of freedom ν							
	1	1.25	1.5	1.75	2	3	4	5
$\text{SNR}(\sigma_{\text{MAD}}) = 40$ dB								
IHT	.51	.86	.95	.99	.99	1.0	1.0	1.0
LIHT	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
HIHT-c_1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
HIHT-c_2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$\text{SNR}(\sigma_{\text{MAD}}) = 20$ dB								
IHT	0	0	0	.01	.04	.31	.57	.72
LIHT	.09	.10	.13	.13	.17	.22	.24	.24
HIHT-c_1	.46	.61	.70	.77	.81	.90	.92	.93
HIHT-c_2	.61	.66	.72	.73	.74	.80	.82	.82

(a)

(b)
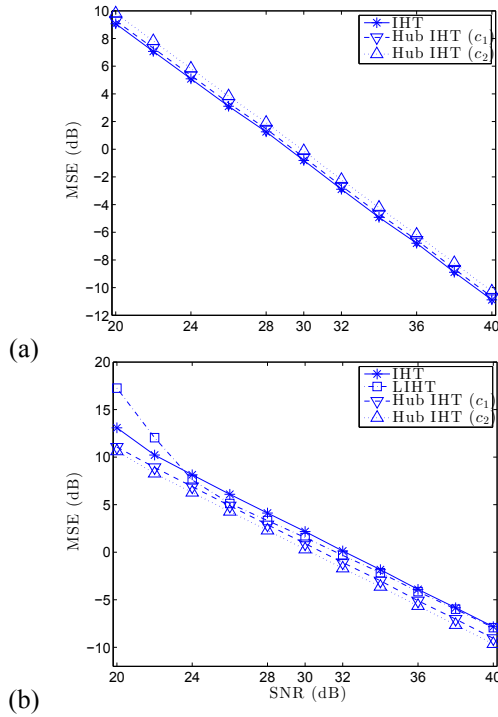
Fig. 1: Average MSE of the methods as a function of $\text{SNR}(\sigma)$ under (a) $\mathcal{N}(0, \sigma_{\text{SD}}^2)$ noise and (b) $Lap(0, \sigma_{\text{MeAD}})$ noise.

only slightly better performance compared to IHT in the high SNR regime [30, 40] dB. The performance loss of IHT as compared to HIHT-c_2 is 1.9 dB in average in SNR [22 40], but jumps to 2.5 dB at SNR 20 dB. Note also that LIHT has the worst performance at low SNR regime. Huber IHT methods and the IHT had a full PER rate ($= 1$) for all SNR in case of Gaussian errors. In case of Laplacian errors, Huber IHT methods had again the best performance and they attained full PER rate at all SNR levels considered. The PER rates of LIHT decayed to 0.9900, 0.9300, and 0.7000 ad SNR = 24, 22, and 20 dB, respectively. The PER rate of IHT was full until SNR 20 dB at which it decayed to 0.97.

Experiment II: Student's t_ν-noise: The noise terms are

(a) $\text{SNR}(\sigma_{\text{MAD}}) = 40$ dB (b) $\text{SNR}(\sigma_{\text{MAD}}) = 20$ dB
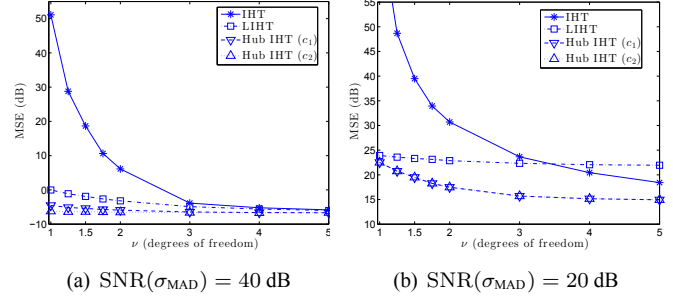
Fig. 2: Average MSE of the methods under $t_\nu(0, \sigma_{\text{MAD}})$ distributed noise as a function of d.o.f. ν.

from Student's t_ν-distribution, $t_\nu(0, \sigma_{\text{MAD}})$. Figure 2(a) and 2(b) depict the MSE as a function of d.o.f. ν at high (40 dB) and low (20 dB) $\text{SNR}(\sigma_{\text{MAD}})$ levels. Huber's IHT methods are outperforming the competing methods in all cases. At high SNR 40dB in Figure 2(a), the Huber IHT with c_2 is able to retain a steady MSE around -6.5 dB for all $\nu \in [1, 5]$. The Huber IHT using c_1 is (as expected) less robust with slightly worse performance, but IHT is already performing poorly at $\nu = 5$ and its performance deteriorates at a rapid rate with decreasing ν. The performance decay of LIHT is much milder than that of IHT, yet it also has a rapid decay when compared to Huber IHT methods. The PER rates given in Table 1 illustrate the remarkable performance of Huber's IHT methods which are able to maintain full recovery rates even at Cauchy distribution (when $\nu = 1$) for SNR 40 dB. At low SNR 20 dB, only the proposed Huber IHT methods are able to maintain good PER rates, whereas the IHT and LIHT provide estimates that are completely corrupted.

5. REFERENCES

[1] M. Elad, *Sparse and redundant representations*, Springer, New York, 2010.

[2] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[3] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: guaranteed stability and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, 2010.

[4] R. E. Carrillo and Barner K. E., "Lorentzian based iterative hard thresholding for compressed sensing," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'11)*, Prague, Czech Republic, May 22 – 27, 2011, pp. 3664 – 3667.

[5] J. L. Paredes and G. R. Arce, "Compressive sensing signal reconstruction by weighted median regression estimates," *IEEE Trans. Signal Processing*, vol. 59, no. 6, pp. 2585–2601, 2011.

[6] S. A. Razavi, E. Ollila, and V. Koivunen, "Robust greedy algorithms for compressed sensing," in *Proc. European Signal Process. Conf.*, Bucharest, Romania, 2012, pp. 969–973.

[7] P. J. Huber, "Robust regression: Asymptotics, conjectures and monte carlo," *Ann. Statist.*, vol. 1, pp. 799–821, 1973.