



HAL
open science

An adaptive robust regression method: Application to galaxy spectrum baseline estimation

Raphael Bacher, Florent Chatelain, Olivier J.J. Michel

► **To cite this version:**

Raphael Bacher, Florent Chatelain, Olivier J.J. Michel. An adaptive robust regression method: Application to galaxy spectrum baseline estimation. ICASSP 2016 - 41st IEEE International Conference on Acoustics, Speech and Signal Processing, Mar 2016, Shanghai, China. pp.4423 - 4427, 10.1109/ICASSP.2016.7472513 . hal-01462974

HAL Id: hal-01462974

<https://hal.science/hal-01462974>

Submitted on 9 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN ADAPTIVE ROBUST REGRESSION METHOD: APPLICATION TO GALAXY SPECTRUM BASELINE ESTIMATION

Raphael Bacher^{*†}, Florent Chatelain[†], Olivier Michel[†]

[†]University Grenoble-Alpes, Gipsa-Lab, Grenoble, France

^{*} CRAL, Observatoire de Lyon, Lyon, France

ABSTRACT

In this paper, a new robust regression method based on the Least Trimmed Squares (LTS) is proposed. The novelty of this approach consists in a simple adaptive estimation of the number of outliers. This method can be applied to baseline estimation, for example to improve the detection of gas spectral signature in astronomical hyperspectral data such as those produced by the new Multi Unit Spectroscopic Explorer (MUSE) instrument. To do so a method following the general idea of the LOWESS algorithm, a classical robust smoothing method, is developed. It consists in a windowed local linear regression, the local regression being done here by the new adaptive LTS approach. The developed method is compared with state-of-the-art baseline estimation algorithms on simulated data closed to the real data produced by the MUSE instrument.

Index Terms— Baseline Estimation, Robust Regression, Least Trimmed Squares, Spectroscopy, Astronomy

1. INTRODUCTION

Classical estimators such as least squares (LS) are not robust to outliers. Thus a number of robust regression approaches have been developed. These methods search to lessen the influence of outliers. Among the most popular approaches are the M-estimators[1]. These methods are based on a penalization of the greatest residuals in the regression process by using a weight function as e.g. the Huber function [1, p.24]. The main advantage of using non truncating weight function, such as the Huber one, is that a global minimization is assured. But the influence of outliers is always present as they are never fully discarded.

Least Trimmed Squares (LTS) [2, p.15] is another classical robust estimator which can be seen as an M-estimator but with a truncating weight function. Indeed it consists in keeping only the smallest residual values and discarding the others. Although this can not assure a global minimization, LTS approach is more robust to outliers as it does not take them into account provided they are correctly discarded. Moreover, in the absence of outlier, this estimator has the same asymptotic properties ($n^{-1/2}$ order of convergence rate) than LS [3, 4].

For all M-estimators a scaling factor is critical. For LTS, this is equivalent to estimating the proportion of outliers. To address this issue, a simple adaptive approach is presented in this paper, based on an empirical distribution of sorted residuals. This results in a new adaptive least trimmed squares (ALTS) which keeps the robustness of a fixed-outliers-proportion LTS while improving the asymptotic efficiency in the absence of outliers.

One application of robust regression is the problem of baseline detection in spectroscopic data. Peaks in a spectrum are considered as "outliers" (there are sparse) and should be separated from a slowly varying baseline.

As explained in [5], a large number of methods have been developed in recent years to deal with baseline estimation in spectroscopy : AIRPLS [6], LOWESS [7], BACKCORR [8] or more recently BEADS [9]. AIRPLS takes a non-parametric Iterative Penalized Least Squares approach. BEADS is based on a filtering approach where the baseline is analog to the low-pass component of the signal. Most of those technics were developed in a context of chemistry and chromatography.

In this paper we place ourselves in a slightly different context. The Multi Unit Spectroscopic Explorer (MUSE) [10] is a new astronomical spectrograph, producing hyperspectral datacube of the deep Universe. New challenges arise such as a low Signal-to-Noise-Ratio (SNR), and the fact that data reduction processes are likely to produce non-positive values in the spectrum. AIRPLS for example is not able to handle the presence of absorption peaks or important symmetric noise. Moreover, the huge number (more than 9×10^4 spectra by datacube) and diversity of spectra to process calls for an unsupervised and adaptive approach. Therefore it is pertinent to derive a robust and adaptive LTS approach. An adaptive LTS was proposed in [11] but it is based on an ad hoc context-oriented criterion, and it can barely be extended to the context of baseline detection.

Section 2 presents the proposed method of adaptive LTS, section 3 explains how to apply this method in a spectral baseline estimation context and in section 4 the proposed method is tested on simulated data close to the MUSE data and compared to state-of-the-art baseline estimation algorithms. Finally some conclusions and perspectives are drawn in section 5.

2. THE ADAPTIVE LTS ALGORITHM

2.1. Least Trimmed Squares Estimation

We are interested in the statistical linear regression problem:

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^n$ is the observation vector, $X \in \mathbb{R}^{n \times q}$ is a given design matrix, $\boldsymbol{\beta} \in \mathbb{R}^q$ is the regression coefficient vector and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a centered white noise vector with variance σ^2 . The classical LS estimator of $\boldsymbol{\beta}$ aims at solving:

$$\hat{\boldsymbol{\beta}}_{\text{LS}} = \arg \min_{\boldsymbol{\beta}} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 = \sum_{i=1}^n [r_i(\boldsymbol{\beta})]^2, \quad (2)$$

where \mathbf{x}_i is the i th column vector of X , while $r_i(\boldsymbol{\beta}) \equiv y_i - \mathbf{x}_i^T \boldsymbol{\beta}$ denotes the i th residual, for $i = 1, \dots, n$.

In a robust framework, we assume now that a proportion of the observations may be atypical data corresponding to outliers. Then, for $i = 1, \dots, n$, the observation model reads $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$ for 'typical' data, while it remains undefined for outliers. By nature, outliers are assumed to be distributed far away from the prediction model $\mathbf{x}_i^T \boldsymbol{\beta}$. This yields large residuals that can spoil the least squares estimator. LTS estimator is a classical robust estimator that discards the largest residuals. For a given $\boldsymbol{\beta}$, let $r_{(i)}$, for $i = 1, \dots, n$, be the ordered absolute values of the residuals such that $|r_{(1)}| \leq |r_{(2)}| \leq \dots \leq |r_{(n)}|$. The α -LTS estimator is defined as [2]

$$\hat{\boldsymbol{\beta}}_{\text{LTS}} = \arg \min_{\boldsymbol{\beta}} \sum_{i=1}^h [r_{(i)}(\boldsymbol{\beta})]^2, \quad (3)$$

where $\alpha \in [1/2, 1]$ and $h = \lceil \alpha n \rceil$ are respectively the proportion and the number of the smallest residuals that are retained in the criterion to be minimized. The definition given in (3) requires to solve a nonconvex optimization problem in order to compute LTS estimator. As explained in [11], this can be viewed as the following extended optimisation problem with constraints:

$$\begin{cases} \min_{\boldsymbol{\beta}, \mathbf{w}} \sum_{i=1}^n w_i (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2, \\ \text{s.t. } \|\mathbf{w}\|_0 \geq h, \mathbf{w} \in \{0, 1\}^n \end{cases} \quad (4)$$

The optimization is now performed jointly on $\boldsymbol{\beta}$ and on the binary weight vector $\mathbf{w} \in \{0, 1\}^n$. This can be solved by alternating two simple steps. For a fixed \mathbf{w} , the global minimizer over $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = (X^T W X)^{-1} X^T W \mathbf{y}, \quad (5)$$

where $W = \text{diag}(\mathbf{w})$ is a diagonal weighting matrix. For a fixed $\boldsymbol{\beta}$, the global minimizer over \mathbf{w} is the binary vector such that, for $i = 1, \dots, n$:

$$w_i = \begin{cases} 1 & \text{if } |r_i(\boldsymbol{\beta})| \leq |r_{(h)}(\boldsymbol{\beta})|, \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

This leads to the iterative algorithm described in Alg. 1. This is

Algorithm 1 LTS

1: *Inputs:* X (design matrix), \mathbf{y} (data), h (number of non-outliers), $\hat{\boldsymbol{\beta}}^0$ (initialization)

2: *Loop:*

3: **while** $\hat{\boldsymbol{\beta}}^k \neq \hat{\boldsymbol{\beta}}^{k-1}$ **do**

4: Update: compute $\hat{\mathbf{w}}^k$ based on $\hat{\boldsymbol{\beta}}_{k-1}$ using (6)

5: Prediction: compute $\hat{\boldsymbol{\beta}}^k$ based on $\hat{\mathbf{w}}^k$ using (5)

6: *Output:* $\hat{\boldsymbol{\beta}}^k$

a classical prediction-correction method, where each step decreases the criterion given in (4). The binary vector \mathbf{w} belonging to a finite space, the convergence to a local minimum of the initial problem (3) is guaranteed after a finite number of iterations. As described in [12], it is possible as in the k-means algorithm to randomly choose several subsets of observations in order to perform several random initializations of the algorithm. This may avoid to be trapped in some local minima, e.g. when the LS estimate (2) is a too bad initial value.

2.2. Adaptive LTS

A classical choice for the trimming proportion parameter of the LTS procedure is given by $\alpha = 1/2$. In this case, the LTS estimator is based on half of the observations having the smallest estimated variance. This leads to a robust estimator with a breakdown point of 50% similar to the median for instance. However the asymptotic efficiency¹ of this estimator is only 7% when there is no outlier [2, p.178-182]. The goal of an adaptive LTS procedure is to estimate the number of non-outliers \hat{h} in order to remain robust, while improving the efficiency of the estimator. Let h_0 and $\pi_0 = h_0/n$ be respectively the true number and the true proportion of non-outliers in the observation vector \mathbf{y} . Consider now the process defined, for $i = 1, \dots, n$, by

$$s_i^2 = \frac{1}{i} \sum_{j=1}^i r_{(j)}^2, \quad (7)$$

where the ordered residuals are obtained from an initial LTS estimator with $\alpha = 1/2$ (the dependence of the residuals over $\boldsymbol{\beta}$ is now omitted for brevity). It is easy to see that $(s_i^2)_{1 \leq i \leq n}$ is a nondecreasing sequence.

2.2.1. Known variance σ^2

When the noise variance σ^2 is known, an estimate \hat{h} of the true number h_0 of non-outliers is obtained as the following stopping time:

$$\hat{h} = \max \{i \in \{1, \dots, n\} : s_i^2 \leq \sigma^2\}. \quad (8)$$

To explain the rationale behind (8), we assume that the residuals associated with the outliers are greater than the h_0 residuals associated with the non-outliers. In this case, the initial LTS procedure with $\alpha = 1/2$ gives a consistent estimate of $\boldsymbol{\beta}$ as long as $h_0 \geq \lceil n/2 \rceil$. Moreover when $i = h_0$, s_i^2 reduces to the sample variance of the h_0 non-outlier residuals. This yields a consistent estimate of the variance σ^2 . Conversely, when $i > h_0$, s_i^2 is an upward biased estimator of σ^2 with a bias that increases rapidly with i as the outlier residuals are greater than the non-outliers one. As a consequence, \hat{h} should be a rather conservative (under) estimate of h_0 .

A new LTS estimation can finally be performed on the \hat{h} smallest residuals. These two steps: 1) estimation \hat{h} of h_0 given in (8) and 2) computation of the resulting LTS estimator, can be repeated until convergence of the estimates, as described in the general case (unknown variance) in Alg. 2.

2.2.2. Unknown variance

In the general case, the noise variance σ^2 is unknown. A first estimation can be obtained on the residuals using the median absolute deviation (MAD) estimate. Based on the Gaussian noise assumption,

$$\hat{\sigma}_{\text{MAD}} = \frac{1}{\Phi^{-1}(3/4)} |r_{(\lceil n/2 \rceil)}|, \quad (9)$$

where Φ^{-1} is the normal quantile function. This estimate can then be plugged in (8) to obtain a first estimate \hat{h} of h_0 . Then an LTS estimation can be performed on the \hat{h} smallest residuals. Moreover, the variance parameter can now be estimated by

$$\hat{\sigma}^2 = \frac{1}{\hat{h}} \sum_{i=1}^{\hat{h}} r_{(i)}^2. \quad (10)$$

¹for a symmetric noise, and in the absence of outlier, the asymptotic efficiency reduces to the ratio of the variance of the least squares estimator divided by the asymptotic variance of the robust estimator

This leads to the iterative algorithm, called Adaptive LTS (ALTS), summarized in Alg. 2.

Algorithm 2 Adaptive LTS (ALTS)

- 1: *Input:* X (design matrix), \mathbf{y} (data), β^0 (initial value)
 - 2: *Initialization:*
 - 3: Compute the $LTS(X, \mathbf{y}, h^0, \beta^0)$ estimator β^1 , with $h^0 = \lceil n/2 \rceil$, where LTS refers to Algo 1
 - 4: Compute $\hat{\sigma}_{MAD}$ using (9)
 - 5: Compute h^1 by plugging $\hat{\sigma}_{MAD}^2$ in (8)
 - 6: *Loop:*
 - 7: **while** $h^k \neq h^{k-1}$ **do**
 - 8: $\hat{\beta}^k = LTS(X, \mathbf{y}, h^{k-1}, \hat{\beta}^{k-1})$
 - 9: Compute $\hat{\sigma}^{2,k}$ using (10)
 - 10: Compute h^k by plugging $\hat{\sigma}^{2,k}$ in (8)
 - 11: *Output:* $\hat{\beta}^k$
-

It is important to note that the proposed ALTS algorithm converges in a finite number of steps. In fact, from (10) and (8), it comes that $\hat{\sigma}^{2,k} = \frac{1}{h_{k-1}} \sum_{i=1}^{h_{k-1}} r_{(i)}^2 \geq s_{h_{k-1}}^{2,k}$. As $h_k = \max \{i \in \{1, \dots, n\} : s_i^{2,k} \leq \hat{\sigma}^{2,k}\}$, it comes that $h_k \geq h_{k-1}$. This shows that $(h_k)_k$ is a nondecreasing integer sequence. This sequence being bounded by the sample size n , it converges in a finite number of steps.

3. APPLICATION TO SPECTRAL FILTERING

One of the main purposes of the wide field spectrograph MUSE [10] is to analyze galaxies and their surroundings by the study of their spectra. Galaxy spectra are composed of a continuum emission and of sparse emission (or absorption) peaks. On the contrary surrounding gas only contains peak such as the Lyman α emission line. To ease the detection of spectral peaks, signature of gas, mixed with galaxies spectral signatures, one approach is to remove the continuous component which can be assimilated to a baseline.

In [7], the LOWESS method was developed to process samples of thousands of values by performing local weighted linear regressions. For time series smoothing, this is equivalent to use a sliding window centered on each sample location in order to estimate the signal at the window center. The samples in the sliding window are weighted according to a tricube function. This function is defined as $f(q) = (1 - |q/l|^3)^3$ where $q \in \{-l, \dots, l\}$ is the relative position in the window of size $2l + 1$. As this local parametric regression is performed for each sample, this yields a non-parametric estimator of the global time series baseline. A robust version of this algorithm consists of iterating this process with new weights obtained by penalizing the highest residuals.

Following the same line of idea as in LOWESS, a weighted sliding-window local regression is used here. However local regression is performed by the ALTS algorithm developed in sec. 2 with a first order polynomial regression. Each point of the signal is therefore estimated by an ALTS approach applied to its weighted neighbours inside the window. A smoothing is finally applied in post-processing by using again local regressions with the tricube weighting function. Note that this estimation procedure depends on a single parameter: the half-window size l .

The complete algorithm is presented in Alg. 3, where $F^{1/2} = \text{diag}(\{f(i)\}_{-l \leq i \leq l})$ denotes the diagonal matrix of the squared root weights given by the tricube function, and $\mathbf{x}_k = (1, t_k)^T \in \mathbb{R}^2$ is the first order polynomial value at time t_k .

Algorithm 3 ALTS Baseline Estimation

- 1: *Input:* \mathbf{y} (data), l (window half size)
 - 2: **for** i in $(1, n)$ **do**
 - 3: $X_i^l = F^{1/2} \times [\mathbf{x}_{i-l}, \dots, \mathbf{x}_{i+l}]^T$
 - 4: $\mathbf{y}_i^l = F^{1/2} \times (y_{i-l}, \dots, y_{i+l})^T$
 - 5: $\beta_i = \text{ALTS}(X_i^l, \mathbf{y}_i^l, \beta_{i-1})$ using Alg. 2
 - 6: $\hat{y}_i = (X_i^l \beta_i)_i$
 - 7: Smooth $\hat{\mathbf{y}}$ using local regressions with a tricube function.
 - 8: *Output:* $\hat{\mathbf{y}}$
-

4. SIMULATION AND RESULTS

4.1. Detection of outliers

We first test algorithm 2 in a non-local framework, with a third order regression. A third order polynomial baseline is built, corrupted by a Gaussian noise and two peaks (with Gaussian shape) corresponding to outliers. The oracle is a least squares regression using the known position of the outliers to discard them.

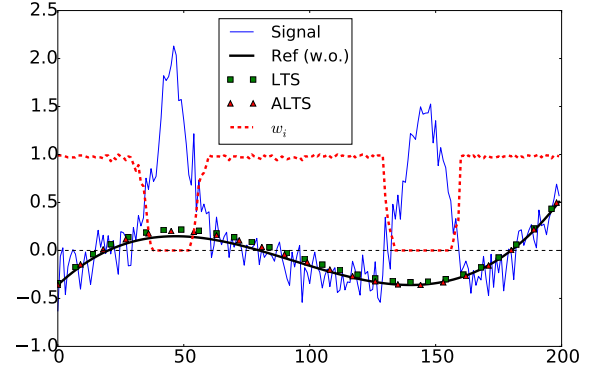


Fig. 1: Mean baseline estimates for 1000 Monte-Carlo runs with $\pi_0 = 0.7$. ALTS estimation (\blacktriangle marker), α -LTS estimation for $\alpha = 0.8$ (\blacksquare marker), one realization of the noisy signal (blue curve), baseline ground-truth (thick black line), mean estimates of the binary weights w_i defined in (4) (red dotted line)

Table 1 shows the results of the baseline estimation for the ALTS algorithm, in the known and unknown variance cases, for the α -LTS algorithm ($\alpha = 0.8$) and for the oracle, with different values of non-outlier proportion π_0 . All the LTS algorithms are initialized with the LS estimate (corrupted by outliers). One sees that π_0 is correctly estimated by ALTS, with and without knowing the variance, for $\pi_0 \geq 0.8$. For $\pi_0 = 0.7$ the number of detected outliers is slightly lower than the real one, even with known variance. This is due to the difficulty to differentiate the tails of the peaks from the noise, as illustrated in Figure 1.

When the trimming factor α in LTS matches the true proportion π_0 of non-outliers, LTS slightly outperforms the adaptive approach. In all other cases it is bested in Mean Integrated Square Error (MISE) by the ALTS. One can also see that as π_0 tends to one, the ALTS reaches the efficiency of the oracle. These results illustrate

that ALTS is both robust to outliers and asymptotically efficient in the absence of outliers.

Table 1: Performances of the ALTS, ALTS with known σ^2 , α -LTS with $\alpha = 0.8$ and oracle for π_0 from 0.7 to 1. Values are integrated over the signal and computed on 1000 Monte Carlo runs

	ALTS	ALTS (known σ^2)	LTS ($\alpha = 0.8$)	Oracle
$\pi_0 = 0.7$				
$\hat{\pi}_0$	0.76	0.73	(0.8)	(0.7)
MISE	0.43	0.36	0.68	0.13
Variance	0.20	0.20	0.17	0.13
Bias ²	0.23	0.16	0.51	1×10^{-5}
$\pi_0 = 0.8$				
$\hat{\pi}_0$	0.81	0.80	(0.8)	(0.8)
MISE	0.21	0.21	0.18	0.15
Variance	0.18	0.18	0.17	0.15
Bias ²	0.025	0.032	0.015	1×10^{-4}
$\pi_0 = 0.9$				
$\hat{\pi}_0$	0.89	0.89	(0.8)	(0.9)
MISE	0.11	0.12	0.18	0.10
Variance	0.11	0.11	0.18	0.10
Bias ²	1.4×10^{-4}	1×10^{-3}	5×10^{-4}	6×10^{-5}
$\pi_0 = 1$				
$\hat{\pi}_0$	0.98	0.98	(0.8)	(1)
MISE	0.09	0.09	0.20	0.087
Variance	0.09	0.09	0.20	0.087
Bias ²	7.4×10^{-5}	9.6×10^{-5}	9.8×10^{-5}	9.6×10^{-5}

4.2. Performance on simulated spectra

A reference spectrum is built using real MUSE data. The baseline ground-truth is built from a spatial average of a galaxy spectrum in MUSE data, filtered by a zero-phase recursive low-pass filter. Two emission lines extracted from real data are added. The parameters used for the additive Gaussian noise those estimated on real data. A realization of the resulting signal is shown in Figure 2 (a).

The proposed method is compared with the BEADS algorithm and the robust version of LOWESS. We tried to find the best parameters for these methods. The cut-off frequency of BEADS was first set to the same value as the parameter used for building the reference spectrum. After testing it appears that the best results were obtained with a slightly lower cut-off frequency. Other parameters were also the best found by trial-error. We choose a common window size for LOWESS and ALTS.

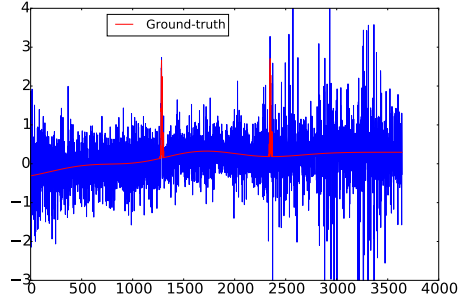
Table 2: ALTS, LOWESS and BEADS. Values are integrated over the signal, or over a 200-window centered on one peak.

	ALTS	LOWESS	BEADS
<i>Overall</i>			
MISE	3.76	4.20	5.84
Variance	2.98	3.26	2.59
Bias ²	0.78	0.94	3.25
<i>Peak zone</i>			
MISE	0.26	0.45	0.91

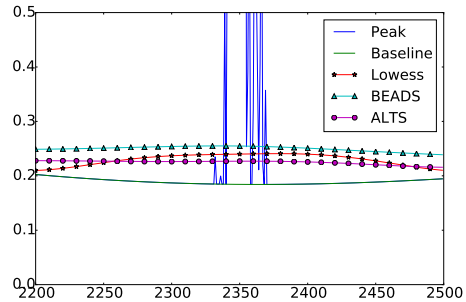
Results in Table 2 are obtained from 200 Monte-Carlo runs. Results from our method appeared to be better than those obtained by the two state-of-the-art methods BEADS and LOWESS, particularly in the peak area as illustrated in Figure 2(b).

4.3. Parameters and computational cost

The final algorithm involves only one parameter: the window size. This later will mostly impact the variance of the estimation since the



(a) Realization of a noisy signal built using MUSE data (blue curve), and noise free signal (red curve)



(b) Baseline estimation around the second peak. Peak (blue curve), baseline (green curve), ALTS estimation (\bullet), LOWESS estimation ($*$), BEADS estimation (\blacktriangle)

Fig. 2: Baseline estimation on simulated data for ALTS, LOWESS, and BEADS averaged on 200 realisations.

larger it is set, the smoother the estimation. Experimental evidence shows that it can be set easily to a reasonable size without deeply degrading the estimation. The computational cost is for now quite high: a few seconds for a 3600 values signal with a Python implementation against 200ms for the MATLAB implementation of BEADS and one second for the python version of LOWESS. Nonetheless, note that BEADS depends on the cut-off frequency plus three penalization weights that must be carefully tuned (especially the generally unknown cut-off parameter).

5. CONCLUSION

In this paper a new robust regression method based on the Least Trimmed Squares estimator was developed. The novelty consists in an adaptive estimation of the number of outliers. This adaptive method can be used in an baseline estimation context by using a local linear regression approach similar to the classical robust estimation algorithm LOWESS. Realistic simulated data mimicking MUSE data were built. Comparison between the proposed method and the state-of-the-art baseline detection algorithms LOWESS and BEADS illustrates the improvement brought by our algorithm but at the price of a higher computational cost.

In some future work the authors would like to explore the convergence properties of the Adaptive Least Trimmed Squares estimator as well as improve the computational costs of the ALTS method. The authors would like to thank the ERC grant 339659-MUSICOS for the funding of this work.

6. REFERENCES

- [1] Ricardo A. Maronna, Douglas Martin, and Victor J. Yohai, *Robust statistics: theory and methods*, Wiley, Chichester, reprinted with corr edition, 2006.
- [2] Peter J. Rousseeuw and Annick M. Leroy, *Robust regression and outlier detection*, Wiley, New York, 1987.
- [3] Jan Ámos Vřšek, “The least trimmed squares. part II: \sqrt{n} -consistency,” *Kybernetika*, vol. 42, no. 2, pp. 181–202, 2006.
- [4] Jan Ámos Vřšek, “The least trimmed squares. part III: Asymptotic normality,” *Kybernetika*, vol. 42, no. 2, pp. 203–224, 2006.
- [5] Komsta, “Comparison of Several Methods of Chromatographic Baseline Removal with a New Approach Based on Quantile Regression,” *Chromatographia*, vol. 73, no. 7-8, pp. 721–731, Apr. 2011.
- [6] Zhi-Min Zhang, Shan Chen, and Yi-Zeng Liang, “Baseline correction using adaptive iteratively reweighted penalized least squares,” *The Analyst*, vol. 135, no. 5, pp. 1138, 2010.
- [7] William S Cleveland, “LOWESS: A program for smoothing scatterplots by robust locally weighted regression,” *American Statistician*, pp. 54–54, 1981.
- [8] Vincent Mazet, David Brie, and Jérôme Idier, “Baseline spectrum estimation using half-quadratic minimization,” in *Signal Processing Conference, 2004 12th European*. 2004, pp. 305–308, IEEE.
- [9] Xiaoran Ning, Ivan W. Selesnick, and Laurent Duval, “Chromatogram baseline estimation and denoising using sparsity (BEADS),” *Chemometrics and Intelligent Laboratory Systems*, vol. 139, pp. 156–167, Dec. 2014.
- [10] R Bacon, M Accardo, L Adjali, H Anwand, S Bauer, I Biswas, J Blaizot, D Boudon, S Brau-Nogue, J Brinchmann, et al., “The MUSE second-generation VLT instrument,” in *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2010, pp. 773508–773508.
- [11] Qianqian Xu, Ming Yan, and Yuan Yao, “Fast Adaptive Least Trimmed Squares for Robust Evaluation of Quality of Experience,” *arXiv preprint arXiv:1407.7636*, 2014.
- [12] Peter J. Rousseeuw and Katrien Van Driessen, “Computing LTS regression for large data sets,” *Data mining and knowledge discovery*, vol. 12, no. 1, pp. 29–45, 2006.