ROBUST CENSORING FOR LINEAR INVERSE PROBLEMS

Georg Kail, Sundeep Prabhakar Chepuri, and Geert Leus

Delft University of Technology (TU Delft), The Netherlands
Email: {g.r.kail; s.p.chepuri; g.j.t.leus}@tudelft.nl.

ABSTRACT

Existing methods for smart data reduction are typically sensitive to outlier data that do not follow postulated data models. We propose robust censoring as a joint approach unifying the concepts of robust learning and data censoring. We focus on linear inverse problems and formulate robust censoring through a sparse sensing operator, which is a non-convex bilinear problem. We propose two solvers, one using alternating descent and the other using Metropolis-Hastings sampling. Although the latter is based on the concept of Bayesian sampling, we avoid confining the outliers to a specific model. Numerical results show that the proposed Metropolis-Hastings sampler outperforms state-of-the-art robust estimators.

Index Terms— Robustness, censoring, sparse sensing, big data.

1. INTRODUCTION

Pervasive sensors, the Internet, and social networks generate massive volumes of data. Such datasets often include redundant and less informative data. Smartly exploiting such redundancy leads to a significant reduction in the data processing costs, since it greatly simplifies solving problems like prediction, estimation, tracking, or classification, to list a few. Thus, the task of extracting the most informative data for further analysis, learning, and inference is of crucial importance. In addition to the data that follow postulated models, the available dataset often includes *outliers* that do not obey them. The presence of such outliers makes data processing tasks significantly more difficult, requiring methods with increased robustness.

Data reduction can be performed before or after acquiring the data, termed *model-driven* or *data-driven* design, respectively. *Sensor selection* [1, 2] determines the most informative data based on a known model before acquiring the data. The best subset of the candidate data is chosen such that a desired ensemble performance is achieved. However, model-driven sensing design schemes inherently lack the ability to reject outliers, as the sensing scheme is agnostic to the data. In *data censoring* [3, 4], less informative data are discarded online. Here, a designed censoring interval determines how informative the data are. However, despite being data-driven, most data censoring schemes are also not designed to be robust to outliers. On the other hand, robust alternatives to least-squares based estimators such as M-estimators [5], least-trimmed-squared (LTS) [6], random sample consensus (RANSAC) [7], or sparsity-controlling outlier rejection [8] are not devised specifically for data censoring.

To this end, we introduce a unifying framework of *robust censoring* for joint robust learning and data censoring. We focus on large-scale linear inverse problems and propose two solvers for the resulting non-convex bilinear problem. The first solver is based on the *alternating descent* method, a standard tool in convex optimization that has low complexity, but yields suboptimal results. The second solver is based on *Metropolis-Hastings sampling* [9], which is here formulated without specifying any prior knowledge about the outliers. Metropolis-Hastings sampling has proven to be an effective method for a wide variety of highly complex estimation problems, especially due to the large flexibility in its formulation. This flexibility, however, requires careful design of the algorithmic steps to ensure convergence within a moderate number of iterations.

2. PROBLEM STATEMENT

Consider a linear regression setup, where an unknown vector $\boldsymbol{\theta} \in \mathbb{R}^N$ is to be estimated from the output data $\{x_m\}_{m=1}^D$ possibly contaminated with up to o outliers. The output data are collected in the vector $\mathbf{x} = [x_1, x_2, \ldots, x_D]^\mathrm{T} \in \mathbb{R}^D$, where $^\mathrm{T}$ denotes transposition. We assume that the acquired data vector $\mathbf{x}$ contains uninformative and/or outlying elements, where we interpret informative data as data that has a large likelihood. The dimensionality of the data is reduced to $d \ll D$ through a linear compression operator $\mathrm{diag_r}(\mathbf{w}) \in \{0,1\}^{d \times D}$ to obtain

$$\mathbf{x}_w = \mathrm{diag_r}(\mathbf{w})\mathbf{x},$$

where $\mathrm{diag_r}(\cdot)$ represents a diagonal matrix with the argument on its diagonal, but with the all-zero rows removed. Here, $w_m = 0$ indicates that x_m is considered outlying or is censored, and $\mathbf{w} = [w_1, w_2, \ldots, w_D]^\mathrm{T}$. The reduced-dimension data vector $\mathbf{x}_w$ is subsequently used to solve the inference or learning problem. The robust censoring problem is stated as follows.

This work was supported in part by the Austrian Science Fund (FWF) under grant J3495 and by NWO-STW under the VICI program (10382).

Problem (Robust Censoring). *Given the data vector $\mathbf{x} \in \mathbb{R}^D$ that is related to the unknown $\boldsymbol{\theta} \in \mathbb{R}^N$ through a known data model but possibly contaminated with up to o outliers: (a) design the Boolean vector $\mathbf{w} \in \{0,1\}^D$ that chooses $d \leq D - o$ data samples discarding possible outliers as well as censoring less-informative samples and (b) use this data to compute an estimate of $\boldsymbol{\theta}$.*

Differently from classical outlier rejection, d may be chosen much smaller than the number of apparently outlier-free data samples. Choosing a smaller d and working only with d-dimensional subvectors of $\mathbf{x}$ often leads to significant reductions of the computational cost. For large D and small d, the postulated $o \leq D-d$ amounts to a very weak assumption about the actual number of outliers. No further assumptions about the outliers are made.

We consider a linear regression problem where the uncontaminated data sample $\bar{x}_m$ is related to the unknown regression coefficients $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^{\mathrm{T}} \in \mathbb{R}^N$ through the following linear model

$$\bar{x}_m = \mathbf{a}_m^{\mathrm{T}} \boldsymbol{\theta} + n_m, \quad m = 1, 2, \ldots, D, \tag{1}$$

where the regressors $\{\mathbf{a}_m\}_{m=1}^D$ collected in the matrix $\mathbf{A} = [\mathbf{a}_1^{\mathrm{T}}, \mathbf{a}_2^{\mathrm{T}}, \ldots, \mathbf{a}_D^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{D \times N}$ are assumed to be known and the noise n_m is Gaussian distributed. To ensure identifiability, we assume that $d \geq N$ and that any N rows of $\mathbf{A}$ are linearly independent, i.e., $\mathbf{A}_w = \mathrm{diag_r}(\mathbf{w})\,\mathbf{A}$ has full column rank for any $\mathbf{w}$ such that $\|\mathbf{w}\|_0 \geq N$.

Given $\mathbf{A}$ and the contaminated data vector $\mathbf{x}$ that potentially contains o outliers, a robust estimator for $\boldsymbol{\theta}$ with respect to the outliers is the well-known LTS estimator

$$\hat{\boldsymbol{\theta}}_{\mathrm{LTS}} = \arg\min_{\boldsymbol{\theta}} \sum_{m=1}^{D-o} r_{[m]}^2(\boldsymbol{\theta}), \tag{2}$$

with the residuals $r_m(\boldsymbol{\theta})$ defined as $r_m(\boldsymbol{\theta}) = x_m - \mathbf{a}_m^{\mathrm{T}} \boldsymbol{\theta}$ and $r_{[m]}^2(\boldsymbol{\theta})$ denoting the squared residuals in ascending order. Furthermore, D–o determines the breakdown point of the LTS as o residuals are not present in (2). The optimization problem (2) incurs combinatorial complexity, where an exhaustive search would include choosing $\hat{\boldsymbol{\theta}}_{\mathrm{LTS}}$ with the smallest cost among all the $\binom{D}{D-o}$ candidate least-squares estimators. For $d = D - o$, the sensing operator $\mathbf{w}$ allows us to recast LTS in (2) as the following optimization problem

$$(\hat{\boldsymbol{\theta}}, \hat{\mathbf{w}}) = \underset{(\boldsymbol{\theta}, \mathbf{w}) \,\in\, \mathbb{R}^N \times \mathcal{W}}{\arg\min} \sum_{m=1}^{D} w_m (x_m - \mathbf{a}_m^{\mathrm{T}} \boldsymbol{\theta})^2, \tag{3}$$

where $\mathcal{W} = \{\mathbf{w} \in \{0,1\}^D \,|\, \|\mathbf{w}\|_0 = d\}$. The above optimization is non-convex in $\mathbf{w}$ and $\boldsymbol{\theta}$ due to the bilinear term, cardinality constraint, and the Boolean constraint on $\mathbf{w}$. The formulation in (3) naturally extends to censoring with $d < D - o$. Various different criteria have been proposed for data censoring, e.g., [4]. For a fixed d, the approach proposed here yields a solution to data censoring that is optimal in the maximum likelihood sense.

We emphasize that while (3) is equivalent to (2) here, the formulation in (3) generalizes to more complicated likelihood functions (e.g., non-Gaussian, non-additive noise models) and observations in correlated noise. Furthermore, the formulation in (3) allows for an extension to hybrid model-and-data-driven designs. These are subjects of ongoing work.

3. PROPOSED SOLVERS

In this section, we derive two solvers for the robust censoring problem. The method presented in Subsection 3.1 is based on classical optimization theory, more specifically the alternating descent technique. This solver is presented here to illustrate the highly multimodal structure of the problem, since its estimates, which correspond to local minima of the cost function, turn out to be frequently incorrect, as shown by numerical results in Section 4. The solver proposed in Subsections 3.2–3.3, which is based on Metropolis-Hastings sampling, on the other hand, does not suffer from this deficiency.

3.1. Alternating descent

The optimization problem (3) can be solved using alternating descent, i.e., alternating minimization with respect to $\boldsymbol{\theta}$ and $\mathbf{w}$. Given $\mathbf{w}$, the cost in (3) is simply a reduced-ordered least-squares problem in the unknown $\boldsymbol{\theta}$, which admits a closed form solution; while given $\boldsymbol{\theta}$, it reduces to a Boolean linear programming problem, which admits an analytical solution with respect to $\mathbf{w}$. These observations suggest an iterative block coordinate descent algorithm yielding successive estimates of $\boldsymbol{\theta}$ with fixed $\mathbf{w}$, and alternately of $\mathbf{w}$ with fixed $\boldsymbol{\theta}$. More specifically, with the iterate of $\mathbf{w}$ given per iteration $i \geq 0$, i.e., $\mathbf{w}[i]$, we solve for $\boldsymbol{\theta}[i]$ using reduced-ordered least-squares as

$$\boldsymbol{\theta}[i] = \boldsymbol{\theta}_{\min}(\mathbf{w}[i]), \tag{4}$$

with

$$\begin{aligned}\boldsymbol{\theta}_{\min}(\mathbf{w}) &= \underset{\boldsymbol{\theta} \,\in\, \mathbb{R}^N}{\arg\min} \;\left\|\mathrm{diag_r}(\mathbf{w})(\mathbf{x} - \mathbf{A}\boldsymbol{\theta})\right\|^2 \\ &= \left(\mathbf{A}_w{}^{\mathrm{T}} \mathbf{A}_w\right)^{-1} \mathbf{A}_w{}^{\mathrm{T}} \mathbf{x}_w\,. \end{aligned} \tag{5}$$

With $\boldsymbol{\theta}[i]$ available, $\mathbf{w}[i+1]$ is obtained as

$$\mathbf{w}[i+1] = \underset{\mathbf{w} \in \mathcal{W}}{\arg\min} \sum_{m=1}^{D} w_m r_m^2(\boldsymbol{\theta}[i])\,.$$

Even though the above linear programming problem has non-convex Boolean and cardinality constraints, there exists a simple analytical solution for $\mathbf{w}[i+1]$ based on ordering the squared residuals $\{r_m^2(\boldsymbol{\theta}[i])\}$. Specifically, the solution $\mathbf{w}[i+1]$ will have entries equal to 1 at indices corresponding to the d smallest squared residuals and zeros otherwise.

The iterations are initialized at $i = 0$ by randomly generating $\mathbf{w}[0]$ from a uniform distribution over $\mathcal{W}$. Note that the alternating descent algorithm converges only to a stationary point of the robust censoring problem (3), and it suffers from the choice of the initial estimate.

3.2. Sample-based estimation

For our formulation of the second proposed solver, let us first rewrite (3) as

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\,\in\,\mathcal{W}}{\arg\min}\ \left\|\mathrm{diag}_{\mathrm{r}}(\mathbf{w})\big(\mathbf{x}-\mathbf{A}\,\boldsymbol{\theta}_{\min}(\mathbf{w})\big)\right\|^2 \tag{6}$$

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{\min}(\hat{\mathbf{w}})\,, \tag{7}$$

with $\boldsymbol{\theta}_{\min}(\mathbf{w})$ as defined in (5). Note that $\hat{\mathbf{w}}$ and $\hat{\boldsymbol{\theta}}$ according to (6) and (7) are still the same as in (3), not approximations as in Subsection 3.1. Inserting (5) into (6) yields

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\,\in\,\mathcal{W}}{\arg\min}\ \left\|\tilde{\mathbf{x}}(\mathbf{w})\right\|^2, \tag{8}$$

with

$$\tilde{\mathbf{x}}(\mathbf{w}) = \left(\mathbf{I}-\mathbf{A}_w\big(\mathbf{A}_w^{\mathrm{T}}\mathbf{A}_w\big)^{-1}\mathbf{A}_w^{\mathrm{T}}\right)\mathbf{x}_w\,.$$

In the following, we focus on finding $\hat{\mathbf{w}}$ according to (8), while $\hat{\boldsymbol{\theta}}$ is simply obtained using (7) and (5).

To overcome the problem of non-convexity of the cost function, we propose to follow the Markov chain Monte Carlo (MCMC) approach [9]. MCMC methods are often employed to find the maximum of some probability distribution, the so-called *target distribution*, which may often be known only up to a normalization constant. Any finite-valued non-negative function of $\mathbf{w} \in \mathcal{W}$ can be interpreted as a non-normalized distribution of $\mathbf{w}$; we can thus use the cost function from (8) as the target distribution $p(\mathbf{w})$ (up to an unknown normalization constant), with slight modifications to turn the minimum into a maximum while preserving non-negativity:

$$p(\mathbf{w}) \propto \exp\left(-\left\|\tilde{\mathbf{x}}(\mathbf{w})\right\|^2\right).$$

We can now write (8) as

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\,\in\,\mathcal{W}}{\arg\max}\ p(\mathbf{w})\,. \tag{9}$$

In accordance with the MCMC concept, we generate a large population of realizations $\mathbf{w}^{(j)}$ from the target distribution $p(\mathbf{w})$ within the domain $\mathcal{W}$. As the number of realizations increases, we can approximate $p(\mathbf{w})$ more and more closely by the sample-based approximation $p_{\mathcal{S}}(\mathbf{w})$, which is defined as the number of realizations $\mathbf{w}^{(j)}$ that are equal to the respective value of $\mathbf{w}$, normalized by the total number of realizations. The sample-based approximation of (9) is then given by $\hat{\mathbf{w}}_{\mathcal{S}} = \arg\max_{\mathbf{w}\,\in\,\mathcal{W}}\ p_{\mathcal{S}}(\mathbf{w})$. However, for moderate sample sizes this approximation has certain known weaknesses (see, e.g., [10, 11] for a more detailed discussion and some alternatives). Instead, we resort to the following widely-used approach (cf. [12]):

$$\hat{\mathbf{w}}_{\mathrm{eval}} = \mathbf{w}^{(j_{\max})} \quad \text{with } j_{\max} = \underset{j}{\arg\max}\ p\big(\mathbf{w}^{(j)}\big)\,, \tag{10}$$

i.e., we evaluate $p\big(\mathbf{w}^{(j)}\big)$ for all j and pick the maximum. This approach is advantageous because it does not require storing the entire population to obtain the global maximum; instead, we can simply compare each new realization $\mathbf{w}^{(j)}$ to the realization that previously maximized $p(\mathbf{w})$. If the new realization achieves a larger $p(\mathbf{w})$, it replaces the previous maximum, otherwise it can be discarded. Thus, throughout the entire process, only one realization needs to be stored. Furthermore, $p(\mathbf{w}^{(j)})$ is already calculated in the process of generating $\mathbf{w}^{(j)}$ and is thus readily available for the comparison. Also note that, in contrast to $\hat{\mathbf{w}}_{\mathcal{S}}$, $\hat{\mathbf{w}}_{\mathrm{eval}}$ can use all realizations including those from the earliest iterations, since transient effects of the initialization do not cause a degradation here.

In principle, (10) would not require that the realizations are generated from $p(\mathbf{w})$. However, generating them from $p(\mathbf{w})$ increases the probability that even a moderate-sized set of realizations contains the maximizer of $p(\mathbf{w})$ within the domain $\mathcal{W}$, which is $\hat{\mathbf{w}}$ according to (9).

3.3. Metropolis-Hastings sampling

For generating the realizations $\mathbf{w}^{(j)}$, we propose to apply Metropolis-Hastings sampling. Each iteration of this algorithm generates—if we ignore for the transient influence of the initialization—a new realization $\mathbf{w}^{(j)}$ from the target distribution $p(\mathbf{w})$. To do this, we first generate a proposal $\widetilde{\mathbf{w}}$ from some proposal distribution $q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)$, whose shape depends on the realization from the previous iteration, i.e., $\mathbf{w}^{(j-1)}$. Then, the new realization $\mathbf{w}^{(j)}$ is chosen as

$$\mathbf{w}^{(j)} = \begin{cases} \widetilde{\mathbf{w}} & \text{with probability } \alpha_j \\ \mathbf{w}^{(j-1)} & \text{with probability } 1-\alpha_j\,, \end{cases} \tag{11}$$

where

$$\alpha_j = \min\left\{\frac{p\big(\widetilde{\mathbf{w}}\big)\,q\big(\mathbf{w}^{(j-1)}\big|\widetilde{\mathbf{w}}\big)}{p\big(\mathbf{w}^{(j-1)}\big)\,q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)}, 1\right\}. \tag{12}$$

There are various ways to determine when to terminate the iterative process, e.g., by examining the distribution of the realizations and judging whether it has converged to a stationary distribution [13]. A simpler approach is to predetermine the number of iterations J based on training.

The proposal distribution $q(\cdot|\cdot)$ is not determined by the Metropolis-Hastings concept but can be chosen freely, under some mild conditions. The choice of $q(\cdot|\cdot)$ is crucial in the sense that it critically determines the rate at which the estimators improve with increasing numbers of iterations. If $q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)$ is concentrated around $\mathbf{w}^{(j-1)}$ too strongly, for example, the estimators typically improve steadily but very slowly, and the algorithm may spend excessive numbers of iterations around local maxima of the target distribution. On the other hand, if $q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)$ is completely independent of $\mathbf{w}^{(j-1)}$, it typically produces excessive numbers of proposals that correspond to small values of α_j and thus fail to improve the estimators because of (11).

Our choice of the proposal procedure and the resulting proposal distribution is as follows. We use two different types of proposals, which we call "small-step" proposals and "large-step" proposals. In each sampler iteration, we decide to make either, with some fixed small probability ρ, a "large-step" proposal or, with probability $1-\rho$, a "small-step" proposal. In our simulations, we set $\rho = 1/(20D)$. We can express $q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)$ as

$$q\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big) = \rho\; q_{\text{large}}\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big) + (1-\rho)\; q_{\text{small}}\big(\widetilde{\mathbf{w}}\big|\mathbf{w}^{(j-1)}\big)\,.$$

For generating a "small-step" proposal $\widetilde{\mathbf{w}}$, we randomly choose two elements from $\mathbf{w}^{(j-1)}$ and flip them. More specifically, $\widetilde{\mathbf{w}}$ is obtained from $\mathbf{w}^{(j-1)}$ by changing the $m^{(\text{add})}$-th element from 0 to 1 and the $m^{(\text{rem})}$-th element from 1 to 0. The index $m^{(\text{add})}$ is chosen using a uniform distribution over all the zero elements of $\mathbf{w}^{(j-1)}$:

$$p_{\text{add}}\big(\,m\,\big|\,\mathbf{w}^{(j-1)}\big) = \frac{1}{D-d}\,, \tag{13}$$

for $m \in \{1,\ldots,D\}$ such that $w_m^{(j-1)} = 0$. For choosing the index $m^{(\text{rem})}$, we assign higher probabilities to indices that correspond to a large entry in the residual $\tilde{\mathbf{x}}(\mathbf{w})$, because removing such indices from the support of $\mathbf{w}$ potentially reduces the cost most (cf. (8)). This choice is based on the same rationale that also underlies the alternating descent algorithm described in Subsection 3.1. The distribution of $m^{(\text{rem})}$ is defined as:

$$p_{\text{rem}}\big(\,m\,\big|\,\mathbf{w}^{(j-1)}\big) = \frac{\big|\,\tilde{x}_m(\mathbf{w}^{(j-1)})\,\big|^2}{\big\|\,\tilde{\mathbf{x}}(\mathbf{w}^{(j-1)})\,\big\|^2}\,, \tag{14}$$

for $m \in \{1,\ldots,D\}$ such that $w_m^{(j-1)} = 1$. The probability of obtaining some $\widetilde{\mathbf{w}}$ from $\mathbf{w}^{(j-1)}$ in a "small-step" proposal is thus

$$q_{\text{small}}(\widetilde{\mathbf{w}}|\mathbf{w}^{(j-1)}) = p_{\text{add}}\big(\,m^{(\text{add})}\,\big|\,\mathbf{w}^{(j-1)}\big) \times p_{\text{rem}}\big(\,m^{(\text{rem})}\,\big|\,\mathbf{w}^{(j-1)}\big)\,,$$

for all $\widetilde{\mathbf{w}}$ that differ from $\mathbf{w}^{(j-1)}$ in exactly one nonzero entry and one zero entry, and 0 for all other $\widetilde{\mathbf{w}}$.

"Large-step" proposals are intended to further reduce the risk of finding only a local maximum of the target distribution. Here, the proposal $\widetilde{\mathbf{w}}$ is chosen from a uniform distribution over $\mathcal{W}$, independently of $\mathbf{w}^{(j-1)}$ or $\mathbf{x}$:

$$q_{\text{large}}(\widetilde{\mathbf{w}}|\mathbf{w}^{(j-1)}) = \frac{1}{\binom{D}{d}}\,. \tag{15}$$

Algorithm 1 MH sampler for robust censoring

```
Initialize with any $\mathbf{w}^{(0)}$ from $\mathcal{W}$,  $\hat{\mathbf{w}}_{\text{eval}} \leftarrow \mathbf{w}^{(0)}$
Iterate for $j = 1,\ldots,J$:
    With probability $\rho$:
        Generate $\widetilde{\mathbf{w}}$ from (15) ("large-step")
    In the converse case:
        Generate $m^{(\text{add})}$ from (13) and $m^{(\text{rem})}$ from (14)
        and calculate $\widetilde{\mathbf{w}}$ ("small-step")
    Calculate $\alpha_j$ according to (12)
    With probability $\alpha_j$:        $\mathbf{w}^{(j)} \leftarrow \widetilde{\mathbf{w}}$
    In the converse case:        $\mathbf{w}^{(j)} \leftarrow \mathbf{w}^{(j-1)}$
    If $p\big(\mathbf{w}^{(j)}\big) > p\big(\hat{\mathbf{w}}_{\text{eval}}\big)$:        $\hat{\mathbf{w}}_{\text{eval}} \leftarrow \mathbf{w}^{(j)}$
```

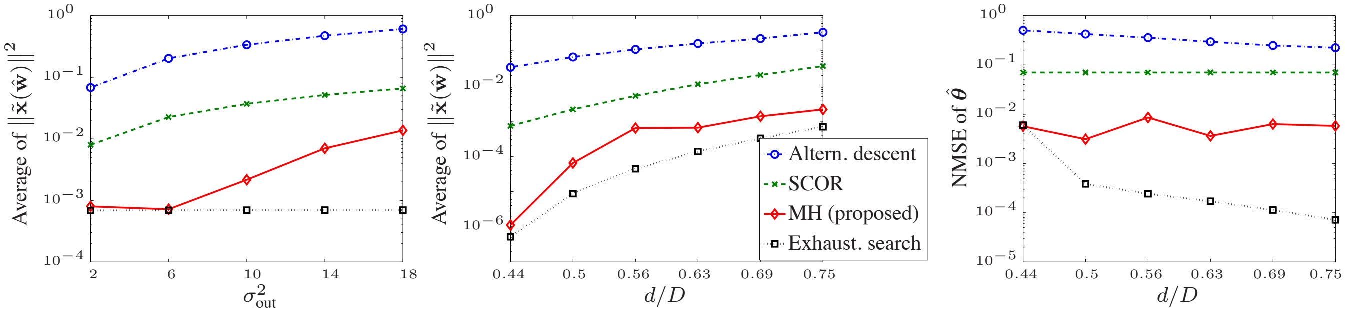
The algorithm is summarized in Algorithm 1. For $d > N$, the complexity per iteration is dominated by $O(Nd^2)$ flops. Note that, in "small-step" iterations, which constitute the vast majority of all iterations, all computations of matrix inverses can be replaced by low-complexity rank-2 updates, since only two elements of $\mathbf{w}$ are changed. This is particularly advantageous for large-scale problems.

4. NUMERICAL EXPERIMENTS

To assess the performance of the proposed method, we generated several thousand data vectors according to (1). We study a setting whose dimensions allow us to perform an exhaustive search, with $D = 16$ and $N = 5$. For each data vector, the elements of $\mathbf{A}$, $\boldsymbol{\theta}$, and $\mathbf{n}$ were individually generated from zero-mean Gaussian distributions with variances $1/N$, 1, and 10^{-4}, respectively. Each row of $\mathbf{A}$ was normalized such that $\|\mathbf{a}_m\| = 1$. In each data vector, $o = 4$ out of the $D = 16$ elements were contaminated with outliers, by adding zero-mean Gaussian noise with variance σ^2_{out}. In different experiments, σ^2_{out} and d were varied to study the behavior of the method in different settings.

For each data vector, $\boldsymbol{\theta}$ and $\mathbf{w}$ were calculated according to Algorithm 1 with $J = 4000$ iterations. The corresponding curves are labeled as MH. As a performance benchmark, we performed an exhaustive search over $\mathcal{W}$ to find $\hat{\mathbf{w}}$ according to (6). Furthermore, we compare the proposed method with the sparsity-controlling outlier rejection [8], which we solve using SeDuMi [14]; we label the corresponding curves as SCOR. The results of the alternating descent method described in Subsection 3.1 are also shown, illustrating its deficiency due to local minima of the cost function. Existing data censoring schemes such as [4, 15] censor measurements with smaller residuals, that is, they do not censor outliers. As a result, the presence of outliers would lead to very poor quality estimates, and thus, we do not compare our results with [4, 15].

Fig. 1(a) shows the average of $\big\|\tilde{\mathbf{x}}(\hat{\mathbf{w}})\big\|^2$, i.e., the cost according to (8) that is achieved by the estimate $\hat{\mathbf{w}}$, using $d = 12$ and different values of σ^2_{out}. We can see that the



(a) (b) (c)

Fig. 1: Estimation performance for different values of σ^2_{out} and d/D: (a) Cost according to (8) achieved by the estimates $\hat{\mathbf{w}}$, for $d/D = 0.75$ and varying σ^2_{out}; (b) The same, for $\sigma^2_{\text{out}} = 10$ and varying d/D; (c) Empirical NMSE of $\hat{\boldsymbol{\theta}}$, for $\sigma^2_{\text{out}} = 10$ and varying d/D; note that our exhaustive search minimizes $\|\tilde{\mathbf{x}}(\hat{\mathbf{w}})\|^2$ rather than the NMSE of $\hat{\boldsymbol{\theta}}$ and is thus not a lower bound in this plot. The shown results are averages over 10 000 experiments.

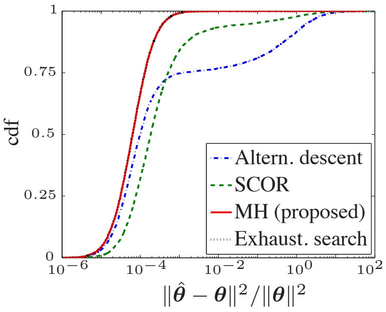


Fig. 2: Empirical cdf of $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2 / \|\boldsymbol{\theta}\|^2$ from 10 000 experiments, with $d/D = 0.75$ and $\sigma^2_{\text{out}} = 10$.

proposed method achieves a significantly lower cost than the other methods, even attaining the optimal results found by exhaustive search for some values of σ^2_{out}. Fig. 1(b) and (c) show results from experiments where σ^2_{out} is fixed at 10 and d is varied between 7 and 12, leading to different compression rates d/D. Fig. 1(b) again shows the average cost according to (8), whereas Fig. 1(c) shows the empirical NMSE of $\hat{\boldsymbol{\theta}}$. As in Fig. 1(a), the proposed method clearly outperforms the other methods in terms of both cost and error. For a more detailed analysis beyond average performance, Fig. 2 shows the empirical cdf of $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2 / \|\boldsymbol{\theta}\|^2$, using the data from Fig. 1(c) at $d = 12$. We can see that the typical performance of all methods is much better than the average one, but some experiments show an exceptionally large error. In the proposed method, however, this effect is less pronounced than in the other methods. Furthermore, its typical performance is better than that of the reference methods, with a cdf that largely coincides with that of exhaustive search.

5. REFERENCES

[1] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[2] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for non-linear measurement models," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 684–698, Feb. 2015.

[3] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: A low-communication-rate scheme for distributed detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 2, pp. 554–568, 1996.

[4] E. J. Msechu and G. B. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 400–414, Jan. 2012.

[5] P. J. Huber, *Robust statistics*, Springer, 2011.

[6] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, vol. 589, John Wiley & Sons, 2005.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[8] J.-J. Fuchs, "An inverse problem approach to robust regression," in *Proc. IEEE ICASSP-1999*, Phoenix, AZ, USA, March 1999, pp. 1809–1812.

[9] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, New York, NY, USA, 2004.

[10] G. Kail, F. Hlawatsch, and C. Novak, "Efficient Bayesian detection of multiple events with a minimum-distance constraint," in *Proc. IEEE SSP-2009*, Cardiff, Wales, UK, Aug.–Sep. 2009, pp. 73–76.

[11] G. Kail, J.-Y. Tourneret, F. Hlawatsch, and N. Dobigeon, "Blind deconvolution of sparse pulse sequences under a minimum distance constraint: A partially collapsed Gibbs sampler method," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2727–2743, June 2012.

[12] N. Dobigeon and J.-Y. Tourneret, "Bayesian orthogonal component analysis for sparse representation," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2675–2685, May 2010.

[13] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statist. Science*, vol. 7, no. 4, pp. 457–472, Nov. 1992.

[14] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[15] G. Wang, D. K. Berberidis, V. Kekatos, and G. B. Giannakis, "Online reconstruction from big data via compressive censoring," in *Proc. GlobalSIP-2014*, Atlanta, GA, USA, Dec. 2014.