On the exact minimization of saturated loss functions for robust regression and subspace estimation

Fabien Lauer

On the exact minimization of saturated loss functions for robust regression and subspace estimation

Fabien Lauer

Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

Abstract

This paper deals with robust regression and subspace estimation and more precisely with the problem of minimizing a saturated loss function. In particular, we focus on computational complexity issues and show that an exact algorithm with polynomial time-complexity with respect to the number of data can be devised for robust regression and subspace estimation. This result is obtained by adopting a classification point of view and relating the problems to the search for a linear model that can approximate the maximal number of points with a given error. Approximate variants of the algorithms based on ramdom sampling are also discussed and experiments show that it offers an accuracy gain over the traditional RANSAC for a similar algorithmic simplicity.

1 Introduction

Robust estimation is a classical problem raised by the presence of outliers in the data. Such outliers are points that do not coincide with the underlying data distribution being learned and that must be rejected in order to estimate an accurate model. A standard approach, entering the statistical framework of redescending M-estimators (Rousseeuw and Leroy, 2005; Shevlyakov et al., 2008), relies on the minimization of a saturated loss function. Indeed, this saturation ensures that outliers yielding gross errors have a very limited influence on the estimation as the gradient of the loss at these points is zero. However, saturated loss functions are inherently nonconvex and their minimization is a highly nontrivial task. For some applications, suboptimal solutions or other heuristics such as the RANdom SAmple Consensus (RANSAC) (Fischler and Bolles, 1981) can provide satisfactory models. Yet, robust estimation problems also appear for instance iteratively in a bounded-error framework for problems where the data is assumed to be generated by a collection of models with unknown assignments of the data points to the models, such as in switching linear regression (Bemporad et al., 2005; Bako, 2011; Lauer, 2016; Lauer and Bloch, 2018) or subspace clustering (Vidal, 2011; Liu et al., 2013; Bako, 2014). In such applications, the models are often estimated one by one while considering the data assigned to other models as outliers. In this context, relying on suboptimal solutions can lead to highly unsatisfactory results with many misclassifications of data points. Robust methods based on convex relaxations (Liu et al., 2013; Bako, 2014; Bako and Ohlsson, 2016) or iteratively hard-thresholding (Bhatia et al., 2015) offer some guarantees but are only optimal under particular conditions on the data.

Instead, in this paper, we aim at unconditional optimality and discuss the computational complexity of globally minimizing a saturated loss function for the robust estimation of linear models, let it be regression ones or subspaces. In particular, the paper focuses on the question of the existence of an algorithm with a polynomial time-complexity with respect to the number of data, N. To this end, we devise an algorithm by enumerating all classifications of the points into two categories: those for which saturation of the loss occurs and those for which it does not. This classification point of view is also motivated by the equivalent formulation of the problem as the maximization of the number of points approximated by a linear model with a bounded error combined with the minimization of a standard (non-saturated) loss over these points only. Indeed, this leads to the

distinction between points with error less than and greater than a predefined threshold. Since there are 2^N binary classifications of N points, such a combinatorial approach based on the enumeration of all of them yields an algorithm with exponential complexity in $\mathcal{O}(2^N)$. Yet, we adopt its classification viewpoint and show that the number of classifications, and thus the complexity, can be reduced to a polynomial function of N. From this classification viewpoint, the minimization of a saturated squared loss for regression can be related to the least trimmed squares estimator (Rousseeuw and Leroy, 2005), for which exact algorithms with polynomial complexity wrt. N have been proposed in Hössjer (1995); Li (2005). However, these are restricted to problems with a single variable (one-dimensional data) and work with a fixed number of inliers rather than an error threshold.

While a polynomial complexity appears convenient, the degree of the polynomials can limit the applicability of the exact algorithms. Therefore, we also briefly discuss approximate variants of the algorithms devised to leverage the computational load by avoiding the complete enumeration of the classifications through random sampling.

Notation We write vectors in lowercase bold letters and matrices in uppercase bold letters. We define $\operatorname{sign}(u)$ as taking value $+1$ if and only if $u \geq 0$ and -1 otherwise. $\operatorname{sign}_0(u)$ is defined similarly except that $\operatorname{sign}_0(0) = 0$. The indicator function $\mathbf{1}_A$ is 1 when the Boolean expression A is true and 0 otherwise.

Paper organization Section 2 gives the precise formulations of the regression and subspace estimation problems we consider. Then, Section 3 shows how these can be solved in polynomial time with respect to N. Section 4 discusses the approximate variants of the algorithms and Section 5 provides a few numerical results. Finally, Section 6 gives concluding remarks.

2 Problem formulation

In general terms, in an estimation problem, one can fit a model to the data by minimizing a loss function of the error between the model output and the data.[1] For instance, standard loss functions include the ℓ_p-losses defined for $p \geq 0$ and all values of the error $e \in \mathbb{R}$ as

$$\ell_p(e) = \begin{cases} \mathbf{1}_{|e|>0}, & \text{if } p = 0 \\ |e|^p, & \text{if } p \in (0, +\infty). \end{cases} \tag{1}$$

Here, we concentrate on robust estimation in the presence of outliers and formulate the problem in terms of a *saturated* loss function $\ell_{p,\epsilon} : \mathbb{R} \to \mathbb{R}^+$, defined for $p \in \{0, 1, 2\}$ by

$$\forall \epsilon > 0, \quad \ell_{p,\epsilon}(e) = \begin{cases} \mathbf{1}_{|e|>\epsilon}, & \text{if } p = 0 \\ (\min(|e|, \epsilon))^p, & \text{if } p \in \{1, 2\}. \end{cases} \tag{2}$$

Indeed, saturating the loss function limits the influence of outliers in the overall cost function to be minimized and thus on the resulting estimate. The statistical properties of these types of loss functions have been studied in the framework of redescending M-estimators, see e.g., Rousseeuw and Leroy (2005). For $p = 0$, this approach is also related to bounded-error estimation. Indeed, we can equivalently view it as the maximization of the number of points for which the error is small and below the threshold ϵ. For $p > 0$, a similar viewpoint can be taken with the additional feature that the small errors are measured by a standard ℓ_p-loss function and further minimized.

In this paper, we will focus the discussion on the corresponding optimization problem whose difficulty comes from the nonconvexity of the saturated losses.

The computation of the argument e as a function of the model parameters and the precise form of the optimization problem depends on the specific problem considered and will be detailed next for regression and subspace estimation.

[1] Note that we focus on problems where the dimensionality is significantly smaller than the number of data and where regularization of linear models might not be necessary. However, given the nature of the proposed approach, introducing a convex regularizer should not raise difficulties.

2.1 Robust regression via saturated loss minimization

The aim of linear regression is to estimate a linear model $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$ from a data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ of regression vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ and target outputs $y_i \in \mathbb{R}$. Here, we adopt an error-minimizing approach and more precisely focus on saturated loss functions as defined above in order to confine the influence of outliers on the global cost. Let us define the index sets $I = \{1, \dots, N\}$ and

$$I_1(\boldsymbol{w}) = \{i \in I : |y_i - \boldsymbol{w}^T\boldsymbol{x}_i| < \epsilon\}, \tag{3}$$

before formally stating the robust regression problem we consider.

Problem 1 ($\ell_{p,\epsilon}$-linear regression)**.** *Given a data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ and a threshold $\epsilon > 0$, find a global solution to*

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \; J_p(\boldsymbol{w}), \tag{4}$$

where

$$\begin{aligned} J_p(\boldsymbol{w}) &= \sum_{i=1}^N \ell_{p,\epsilon}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i) \qquad (5)\\ &= \begin{cases} N - |I_1(\boldsymbol{w})|, & \text{if } p = 0 \\ \displaystyle\sum_{i\in I_1(\boldsymbol{w})} |y_i - \boldsymbol{w}^T\boldsymbol{x}_i|^p + \epsilon^p(N - |I_1(\boldsymbol{w})|), & \text{if } p \in \{1, 2\}\,. \end{cases} \end{aligned}$$

The formulation of Problem 1 emphasizes the connection between saturated loss minimization and bounded-error estimation, i.e., the maximization of the number of points approximated with a bounded error that are here marked with index in $I_1(\boldsymbol{w})$.

This also draws a connection with the classification problem of separating between points that are approximated with a bounded error by an optimal model and those that are not. In particular, given the solution to this classification problem, i.e., $I_1(\boldsymbol{w}^*)$ for some global minimizer $\boldsymbol{w}^*$ of $J_p(\boldsymbol{w})$, a (perhaps different[2]) global solution $\hat{\boldsymbol{w}}$ can be recovered by solving Problem 1 under the constraint $I_1(\boldsymbol{w}) = I_1(\boldsymbol{w}^*)$. Then, for $p = 0$, $J_p(\boldsymbol{w})$ is a mere constant and it suffices to find a $\boldsymbol{w}$ such that $|y_i - \boldsymbol{w}^T\boldsymbol{x}_i| < \epsilon$ for all $i \in I_1(\boldsymbol{w}^*)$ to satisfy the constraint. Conversely, for other values of p, the cost function $J_p(\boldsymbol{w})$ simplifies to a sum of error terms over a fixed set of points plus a constant. Hence, its minimization amounts to a standard regression problem with a non-saturated loss and we can compute $\hat{\boldsymbol{w}}$ by solving

$$\hat{\boldsymbol{w}} \in \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \begin{cases} \displaystyle\max_{i\in I_1(\boldsymbol{w}^*)} |y_i - \boldsymbol{w}^T\boldsymbol{x}_i|, & \text{if } p = 0 \\ \displaystyle\sum_{i\in I_1(\boldsymbol{w}^*)} |y_i - \boldsymbol{w}^T\boldsymbol{x}_i|^p, & \text{otherwise.} \end{cases} \tag{6}$$

Such standard problems have polynomial complexities in $\mathcal{O}(d^2N)$ for $p = 2$ and $\mathcal{O}(d^4N^4)$ for $p \in \{0, 1\}$.

2.2 Robust subspace estimation via saturated loss minimization

A d_s-dimensional subspace of $\mathbb{R}^d$ can be thought of as the column space of a $d \times d_s$ matrix $\boldsymbol{B}$ with orthonormal columns. In this case, the projection of a vector $\boldsymbol{x} \in \mathbb{R}^d$ onto the subspace can be written as $\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{x}$ and the corresponding scalar approximation error as $\|(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}\|$.

Therefore, subspace estimation from a data set $\{\boldsymbol{x}_i\}_{i=1}^N$ with a fixed subspace dimension equal to d_s can be set as the search for a matrix $\boldsymbol{B} \in \mathbb{R}^{d\times d_s}$ such that $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}$ and that the approximation error is minimized over the data set. In the presence of outliers, a robust estimation can be obtained from the minimization of a saturated loss function (as defined in (2)) of this approximation error.

For any $\boldsymbol{B} \in \mathbb{R}^{d\times d_s}$, we define the index set

$$I_1(\boldsymbol{B}) = \{i \in I : \|(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\| < \epsilon\}, \tag{7}$$

in order to state the problem of robust subspace estimation as follows.

[2]Problem 1 may have multiple global solutions, especially when $p = 0$.

Problem 2 ($\ell_{p,\epsilon}$-subspace estimation)**.** *Given a data set* $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^d$*, a subspace dimension* d_s *and a threshold* $\epsilon > 0$*, find a global solution to*

$$\min_{\boldsymbol{B}\in\mathbb{R}^{d\times d_s}} \; J_p^S(\boldsymbol{B}), \quad s.t.\ \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}, \tag{8}$$

where

$$\begin{aligned} J_p^S(\boldsymbol{B}) &= \sum_{i=1}^N \ell_{p,\epsilon}\left(\|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\|\right) \\ &= \begin{cases} N - |I_1(\boldsymbol{B})|, & \text{if } p = 0 \\ \displaystyle\sum_{i\in I_1(\boldsymbol{B})} \|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\|^p + \epsilon^p(N - |I_1(\boldsymbol{B})|), & \text{if } p \in \{1,2\}. \end{cases} \end{aligned} \tag{9}$$

As for robust regression, our formulation emphasizes the classification point of view: if the classification of the point indexes into $I_1(\boldsymbol{B}^*)$ was known for some optimal $\boldsymbol{B}^*$, a (possibly different) global solution $\hat{\boldsymbol{B}}$ to Problem 2 could be obtained by solving a more simple subspace estimation subproblem of the form

$$\begin{aligned} \hat{\boldsymbol{B}} \in \operatorname*{argmin}_{\boldsymbol{B}\in\mathbb{R}^{d\times d_s}} & \begin{cases} \displaystyle\max_{i\in I_1(\boldsymbol{B}^*)} \|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\|, & \text{if } p = 0 \\ \displaystyle\sum_{i\in I_1(\boldsymbol{B}^*)} \|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\|^p, & \text{otherwise} \end{cases} \\ s.t.\ & \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}. \end{aligned} \tag{10}$$

For instance, for $p = 2$, the solution to (10) is computable in $\mathcal{O}(d^2N)$ time via the singular value decomposition of the matrix $\boldsymbol{X}_1$ made of the data points $\boldsymbol{x}_i$ with index $i \in I_1(\boldsymbol{B}^*)$ as columns, $\boldsymbol{X}_1 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, by extracting a subset of columns $\boldsymbol{u}_k$ from $\boldsymbol{U}$: $\hat{\boldsymbol{B}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{d_s}]$.

3 Exact algorithms with polynomial time-complexity with respect to N

We now turn to the analysis of the computational complexity of Problems 1–2 wrt. N, i.e., for a fixed data dimension d. In particular, we will show that, under simple assumptions on the genericity of the point distributions, these complexities are no more than polynomial.

Assumption 1. *In Problem 1, the points* $\{[y_i, \boldsymbol{x}_i^T]^T\}_{i=1}^N \cup \{\boldsymbol{0}\}$ *are in general position, i.e., no hyperplane passing through the origin of* $\mathbb{R}^{d+1}$ *contains more than* d *points from* $\{[y_i, \boldsymbol{x}_i^T]^T\}_{i=1}^N$*.*

Assumption 2. *In Problem 2, the points* $\{\boldsymbol{x}_i\}_{i=1}^N$ *are in general position, i.e., no hyperplane of* $\mathbb{R}^d$ *contains more than* d *of these points.*

Sections 3.1 and 3.2 will prove the existence of exact algorithms that run in polynomial time for Problems 1 and 2. In both cases, these algorithms will be devised with the following approach. As discussed above, thanks to the formulations (5) and (9) expressing the objective functions in terms of the index sets $I_1(\boldsymbol{w})$ and $I_1(\boldsymbol{B})$ of points with error less than ϵ, we can compute an optimal solution $\hat{\boldsymbol{w}}$ or $\hat{\boldsymbol{B}}$ from the knowledge of the optimal set $I_1^* = I_1(\boldsymbol{w}^*)$ or $I_1^* = I_1(\boldsymbol{B}^*)$. Thus, the algorithm can perform a combinatorial search for the index set I_1^* rather than a continuous optimization over $\boldsymbol{w}$ or $\boldsymbol{B}$. The number of sets $I_1 \subseteq I$ being finite, we can enumerate them and compute the optimal continuous variables and objective function values for each one of them with the guarantee of finding a global solution. The main difficulty with this approach is that the number of sets $I_1 \subseteq I$ is 2^N and thus exponential in N. However, we will show below that the sets $I_1(\boldsymbol{w})$ and $I_1(\boldsymbol{B})$ can be obtained via linear classification for any $\boldsymbol{w}$ and $\boldsymbol{B}$ and that all the corresponding linear classifications can be enumerated in polynomial time wrt. N.

This reduction to a polynomial complexity is based on recent results from Lauer (2015), where it is shown that the number of hyperplanes producing different classifications of N points is on the

order of $\mathcal{O}(N^d)$ in $\mathbb{R}^d$ and that the complexity of constructing these hyperplanes is of a similar order, i.e., all hyperplanes can be computed in $\mathcal{O}(N^d)$ operations.[3] More precisely, we will need an adaptation of these results, stated as Proposition 1 below, in order to deal with linear instead of affine classifiers and to work under less restrictive conditions on the distribution of the points. Indeed, Proposition 3 in Lauer (2015) requires the points to be in general position to avoid having too many (i.e., a number proportional to N) undetermined classifications of points lying exactly on the separating hyperplane. In our framework below, we apply this proposition not to data points but to their projection in a space where this assumption cannot hold. Hence, we will instead prove that the number of projected points falling on the hyperplane cannot exceed a certain constant if the original data points are in general position and build the algorithm to explicitly deal with these points.

Proposition 1. *For any binary linear classifier $h(\boldsymbol{x}) = sign(\boldsymbol{h}^T\boldsymbol{x})$, $\boldsymbol{h} \in \mathbb{R}^d$, and any finite set of $N \geq d$ points $S = \{\boldsymbol{x}_i\}_{i=1}^N$, there is a subset of points $S_h \subset S$ of cardinality $|S_h| = d-1$ and a separating hyperplane of normal $\boldsymbol{h}_{S_h}$ passing through the points in S_h, i.e.,*

$$\forall \boldsymbol{x} \in S_h, \quad \boldsymbol{h}_{S_h}^T \boldsymbol{x} = 0, \quad \text{with } \|\boldsymbol{h}_{S_h}\| = 1, \tag{11}$$

which yields the same classification of S in the sense that

$$\forall \boldsymbol{x}_i \in S \setminus \{\boldsymbol{x} \in S : \boldsymbol{h}_{S_h}^T \boldsymbol{x} = 0\}, \quad h(\boldsymbol{x}_i) = sign(\boldsymbol{h}_{S_h}^T \boldsymbol{x}_i). \tag{12}$$

Note that the set $\{\boldsymbol{x} \in S : \boldsymbol{h}_{S_h}^T \boldsymbol{x} = 0\}$ includes S_h but can also include more than $d-1$ points if S is not in general position.

Proof sketch. The general sketch of the proof is similar to the one of Proposition 3 in Lauer (2015): the hyperplane of normal $\boldsymbol{h}$ is transformed to the one of normal $\boldsymbol{h}_{S_h}$ via a series of translation and rotations, while making sure that the classification remains unchanged except for the points that end up lying on the hyperplane. The first difference is that the set of classifiers is restricted to linear instead of affine ones by only considering hyperplanes passing through the origin, and thus with one less degree of freedom and one less point to choose in S_h to fix the hyperplane. Technically, this removes the need for the first translation and only rotations are used to transform the hyperplane. Another difference is that, due to the lack of assumption on the distribution of the points, the number of additional points through which the hyperplane passes after a rotation is unbounded (otherwise than by N). Thus, the statement in (12) explicitly excludes all points on the hyperplane and appears slightly weaker than the one in Lauer (2015) in that it might apply to less points. □

3.1 $\ell_{p,\epsilon}$-linear regression

We first focus on Problem 1 and start with a classification-based reformulation of the set of indexes $I_1(\boldsymbol{w}^*)$ of points with error smaller than ϵ. This relies on the construction of a classification data set

$$\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{2N}, \text{ with } \boldsymbol{z}_i = \begin{cases} [y_i - \epsilon, \ -\boldsymbol{x}_i^T]^T, & \text{if } i \leq N \\ [-y_{i-N} - \epsilon, \ \boldsymbol{x}_{i-N}^T]^T, & \text{if } i > N \end{cases}. \tag{13}$$

Lemma 1. *Given a parameter vector $\boldsymbol{w} \in \mathbb{R}^d$, the set $I_1(\boldsymbol{w})$ defined in (3) is given by*

$$I_1(\boldsymbol{w}) = \left\{ i \in I : q_i = q_{i+N} = -1, \quad q_i = sign\left(\boldsymbol{h}(\boldsymbol{w})^T \boldsymbol{z}_i\right) \right\}, \tag{14}$$

where $\boldsymbol{h}(\boldsymbol{w}) = [1, \boldsymbol{w}^T]^T$ and $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{2N}$ as in (13).

Proof. For any $i \in I$, we have $i \leq N$ and $q_i = \text{sign}(\boldsymbol{h}(\boldsymbol{w})^T \boldsymbol{z}_i) = \text{sign}(y_i - \epsilon - \boldsymbol{w}^T \boldsymbol{x}_i)$, while $q_{i+N} = \text{sign}(\boldsymbol{h}(\boldsymbol{w})^T \boldsymbol{z}_{i+N}) = \text{sign}(-y_i - \epsilon + \boldsymbol{w}^T \boldsymbol{x}_i)$. Thus,

$$\begin{aligned} |y_i - \boldsymbol{w}^T \boldsymbol{x}_i| < \epsilon &\Leftrightarrow y_i - \epsilon - \boldsymbol{w}^T \boldsymbol{x}_i < 0 \wedge -y_i - \epsilon + \boldsymbol{w}^T \boldsymbol{x}_i < 0 \\ &\Leftrightarrow q_i = -1 \wedge q_{i+N} = -1 \end{aligned}$$

and recalling the definition of $I_1(\boldsymbol{w})$ in (3) completes the proof. □

[3] Note that a polynomial bound on the number of linear classifications of N points such as the one obtained with the celebrated Sauer's lemma and Vapnik-Chervonenkis dimension of hyperplanes is not sufficient for our purposes, since it does not provide an algorithm to explicitly enumerate the classifications.

We will also need the following.

Lemma 2. *Given a data set* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ *satisfying Assumption 1, no hyperplane of* $\mathbb{R}^{d+1}$ *passing through the origin can pass through more than* d *points of each of the sets* $\mathcal{Z}_1 = \{\boldsymbol{z}_i = [y_i - \epsilon,\ -\boldsymbol{x}_i^T]^T\}_{i=1}^N$, $\mathcal{Z}_2 = \{\boldsymbol{z}_i = [-y_{i-N} - \epsilon,\ \boldsymbol{x}_{i-N}^T]^T\}_{i=N+1}^{2N}$ *and thus through more than* $2d$ *points of* $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$.

Proof. By Assumption 1, each of the sets $\mathcal{Z}_1 \cup \{\mathbf{0}\}$ and $\mathcal{Z}_2 \cup \{\mathbf{0}\}$ is also in general position. Thus, no hyperplane of $\mathbb{R}^{d+1}$ can pass through more than d points in each of these sets, and hence through more than a total of $n \leq 2d$ points of $\mathcal{Z}$. □

Using the classification viewpoint of Lemma 1, we can state the following result which considers the case where more than d data points can be approximated with error less than ϵ by a linear model. Note that there are always at least d such points and that the case where there are precisely d is trivial since any group of d points yields an optimal solution.

Proposition 2. *Assume that the global minimum of Problem 1 is* $J_p^* < \epsilon^p(N-d)$. *Then, under Assumption 1, in* $\mathbb{R}^{d+1}$, *there is a hyperplane of normal* $\boldsymbol{h} \in \mathbb{R}^{d+1}$ *passing through the origin and* $n \in [d, 2d]$ *points of* $\mathcal{Z}$ *as in (13) such that* $h_1 > 0$ *and*

i) a global solution can be computed by solving $2^n \leq 2^{2d}$ *standard subproblems (6) that can be built from* $\boldsymbol{h}$,

ii) for $p = 0$, $\boldsymbol{w} = \boldsymbol{h}_{2:d+1}/h_1$ *is an approximate solution with* $J_0(\boldsymbol{w}) \leq J_0^* + n \leq J_0^* + 2d$.

Proof. First note that, as a direct consequence of Lemma 2, we always have $n \leq 2d$.

Part i) Let $\mathcal{W}_p^*$ be the set of global minimizers of Problem 1. By Lemma 1, for all $\boldsymbol{w}^* \in \mathcal{W}_p^*$ there is a hyperplane of normal $\boldsymbol{h}(\boldsymbol{w}^*)$ classifying the $2N$ points of $\mathcal{Z}$ into $I_1(\boldsymbol{w}^*)$ and $I \setminus I_1(\boldsymbol{w}^*)$. By Proposition 1, there is an equivalent hyperplane of normal $\boldsymbol{h}^*$ passing through $n \geq d$ points of $\mathbb{R}^{d+1}$, such that

$$\forall i \in \{1, \ldots, 2N\} \setminus I_0, \quad \operatorname{sign}\left(\boldsymbol{z}_i^T \boldsymbol{h}(\boldsymbol{w}^*)\right) = \operatorname{sign}\left(\boldsymbol{z}_i^T \boldsymbol{h}^*\right), \tag{15}$$

where $I_0 = \{i : \boldsymbol{z}_i^T \boldsymbol{h}^* = 0\}$. Thus, the set $I_1(\boldsymbol{h}_{2:d+1}^*/h_1^*)$ computed as in Lemma 1 differs from $I_1(\boldsymbol{w}^*)$ by at most $n = |I_0|$ entries and $I_1(\boldsymbol{w}^*)$ must be one of the 2^n sets $I_1^{\boldsymbol{s}} = \{i \in I : q_i = q_{i+N} = -1, \quad q_i = \operatorname{sign}(\boldsymbol{h}(\boldsymbol{w})^T \boldsymbol{z}_i),\ \text{if } i \notin I_0,\ q_i = q_{i_k} = s_k,\ \text{otherwise}\}$, where we indexed the entries in I_0 as $I_0 = \{i_1, \ldots, i_n\}$ and $\boldsymbol{s} \in \{-1, +1\}^n$ encodes the classification of the corresponding points. Solving the subproblem (6) over the data points with index in $I_1^{\boldsymbol{s}}$ for all $\boldsymbol{s} \in \{-1, +1\}^n$ then yields at least one global solution in $\mathcal{W}_p^*$.

Given that $J_p^* < \epsilon^p(N-d)$, we have $|I_1(\boldsymbol{w}^*)| > d$ for all $\boldsymbol{w}^* \in \mathcal{W}_p^*$, which implies $h_1^* > 0$ as follows. First, note that $I_1(\boldsymbol{w}^*) \setminus I_0$ is not empty: by Lemma 2, I_0 contains no more than d indexes within I while $I_1(\boldsymbol{w}^*) \subseteq I$ and $|I_1(\boldsymbol{w}^*)| > d$. Then, by Lemma 1 and the sign equalities (15), for some $i \in I_1(\boldsymbol{w}^*) \setminus I_0$, $\boldsymbol{z}_i^T \boldsymbol{h}^* < 0$ and $\boldsymbol{z}_{i+N}^T \boldsymbol{h}^* < 0$. Assume $h_1^* = 0$, then $\boldsymbol{z}_i^T \boldsymbol{h}^* = -\boldsymbol{x}_i^T \boldsymbol{h}_{2:d+1}^* = -\boldsymbol{z}_{i+N}^T \boldsymbol{h}^*$, which shows a contradiction with the previous statement. Similarly, letting $h_1^* < 0$ and using $\boldsymbol{z}_i^T \boldsymbol{h}^* < 0$ yields $\boldsymbol{z}_{i+N}^T \boldsymbol{h}^* = -\boldsymbol{z}_i^T \boldsymbol{h}^* - 2\epsilon h_1^* > -2\epsilon h_1^* > 0$, and a contradiction with $\boldsymbol{z}_{i+N}^T \boldsymbol{h}^* < 0$. Thus, $h_1^* > 0$.

Part ii) For $p = 0$, given $h_1^* > 0$, $\operatorname{sign}(\boldsymbol{z}_i^T \boldsymbol{h}^*) = \operatorname{sign}(\boldsymbol{z}_i^T \boldsymbol{h}^*/h_1^*)$. Thus, if $\boldsymbol{s}^*$ is a choice of $\boldsymbol{s}$ yielding an optimal solution with $\boldsymbol{s}_k^* = \operatorname{sign}(\boldsymbol{z}_{i_k}^T \boldsymbol{h}^*)$, $k = 1, \ldots, n$, by Lemma 1, $I_1(\boldsymbol{w}^*) = I_1^{\boldsymbol{s}^*} = I_1(\boldsymbol{h}_{2:d+1}^*/h_1^*)$. Given that for $p = 0$ any $\boldsymbol{w}$ such that $I_1(\boldsymbol{w}) = I_1(\boldsymbol{w}^*)$ yields the same cost $J_0(\boldsymbol{w}) = J_0(\boldsymbol{w}^*)$, we conclude that in this case $\boldsymbol{w} = \boldsymbol{h}_{2:d+1}^*/h_1^* \in \mathcal{W}_0^*$. Since $\boldsymbol{s}_k^* \neq \operatorname{sign}(\boldsymbol{z}_{i_k}^T \boldsymbol{h}^*)$ only occurs for at most n values of k, $I_1(\boldsymbol{w})$ deviates by at most n entries from this case and $|I_1(\boldsymbol{w})| \leq |I_1(\boldsymbol{w}^*)| + n$, yielding the second statement. □

Proposition 2 shows that a global minimizer of Problem 1 can be obtained from a particular separating hyperplane of $\mathcal{Z}$. In addition, this hyperplane can be built from a subset of $\mathcal{Z}$ of cardinality $n \in [d, 2d]$ as the one that passes through the origin of $\mathbb{R}^{d+1}$ and the n points in the subset. Since any subset of d points among these n points yields the same hyperplane, it suffices to find one particular subset of d points among $\mathcal{Z}$. Hence, the problem is reduced to a combinatorial search with $\binom{2N}{d}$ main iterations.

This is formally stated in Algorithm 1 and the theorem below.

Theorem 1. *Under Assumption 1 and given that subproblem (6) can be solved in $\mathcal{O}(N^c)$ time with a constant $c \geq 1$ independent of N, Algorithm 1 solves Problem 1 in $\mathcal{O}(N^{c+d})$ operations.*

Proof. For all subsets of d points of $\mathcal{Z}$, Algorithm 1 computes the hyperplane passing through these points and of orientation such that $h_1 > 0$. By Proposition 2, at least one of these hyperplanes with normal $\boldsymbol{h}$ is such that the inner loop over $\boldsymbol{s}$ in Algorithm 1 finds a global solution.

The computational complexity of Algorithm 1 is the number of subsets of d points among $2N$ times the time needed for a single iteration, which includes computing a normal vector $\boldsymbol{h}$, classifying $\mathcal{Z}$ with $\boldsymbol{h}$, the inner loop over $\boldsymbol{s}$ and the subproblem (6). The normal $\boldsymbol{h}$ of a hyperplane passing through the origin and d points $\{\boldsymbol{z}_{i_k}\}_{k=1}^d$ in $\mathbb{R}^{d+1}$ can be computed as a unit vector in the null space of $[\boldsymbol{z}_{i_1}, \dots, \boldsymbol{z}_{i_d}]^T$, extracted in $\mathcal{O}(d^3)$. If $h_1 < 0$, then $-\boldsymbol{h}$ is a normal for the hyperplane of opposite orientation. Given that $\boldsymbol{z}_{i+N}^T \boldsymbol{h} = -\boldsymbol{z}_i^T \boldsymbol{h} - 2\epsilon h_1$, the classification step takes $\mathcal{O}(N(d+2))$ instead of $\mathcal{O}(2N(d+1))$. The inner loop contains $2^n \leq 2^{2d}$ iterations with $\mathcal{O}(N^c)$ operations each to solve an instance of (6) over at most N data points. Overall, we obtain a time complexity of

$$\begin{aligned} T(N) &= \mathcal{O}\left(\binom{2N}{d}\left(d^3 + (d+2)N + 2^{2d}N^c\right)\right) \\ &= \mathcal{O}\left(\frac{N^d}{d!}(d^3 + (d+2)N + 2^{2d}N^c)\right) \\ &= \mathcal{O}\left(N^{d+c}\right). \end{aligned}$$

□

As an example, applying Theorem 1 for the saturated square loss, $\ell_{2,\epsilon}$, yields the exact Algorithm 1 for robust regression with complexity in the order of $\mathcal{O}(N^{d+1})$.

Algorithm 1 Exact $\ell_{p,\epsilon}$-regression

Input: a data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, a threshold $\epsilon > 0$.
  Initialize $J^* \leftarrow \epsilon^p N$.
  for all $S \subset \mathcal{Z}$ with cardinality $|S| = d$ **do**
    Compute the normal $\boldsymbol{h}$ to the hyperplane passing through $S \cup \{\boldsymbol{0}\}$ with orientation such that $h_1 \geq 0$.
    if $h_1 \neq 0$ **then**
      Classify the points:
      $\forall i \in \{1, \dots, 2N\},\ q_i = \text{sign}_0(\boldsymbol{h}^T \boldsymbol{z}_i)$.
      Set $I_0 = \{i \in \{1, \dots, 2N\} : q_i = 0\}, \quad n = |I_0|$.
      for all $\boldsymbol{s} \in \{-1, +1\}^n$ **do**
        Set the entries of $\boldsymbol{q}$ with index in I_0 to $\boldsymbol{s}$
        Compute $I_1^{\boldsymbol{s}} = \{i \leq N : q_i = q_{i+N} = -1\}$
        if $\epsilon^p(N - |I_1^{\boldsymbol{s}}|) < J^*$ **then**
          Compute $\hat{\boldsymbol{w}}$ as in (6) with $I_1^{\boldsymbol{s}}$ replacing $I_1(\boldsymbol{w}^*)$.
          if $J_p(\hat{\boldsymbol{w}}) < J^*$ **then**
            Update $J^* \leftarrow J_p(\hat{\boldsymbol{w}}),\ \boldsymbol{w}^* \leftarrow \hat{\boldsymbol{w}}$
          end if
        end if
      end for
    end if
  end for
  return $J^*, \boldsymbol{w}^*$

Remark Note that a number of details can be included in Algorithm 1 to make it more efficient in practice. First, it can be easily parallelized. Then, a number of computations can be spared. For instance, since the changes in $\boldsymbol{s}$ can only incur a difference of at most n on $|I_1^{\boldsymbol{s}}|$, the inner loop over $\boldsymbol{s}$ can be skipped when I_1 computed with $q_i = \text{sign}(\boldsymbol{h}^T \boldsymbol{z}_i)$ is such that $\epsilon^p(N - |I_1|) > J_p^* + \epsilon^p n$. Also, for $p = 0$, since $J_p(\boldsymbol{w}) = N - |I_1^{\boldsymbol{s}}|$ is constant for a fixed classification, we do not need to solve (6) to update J^* and $\boldsymbol{w}^*$ can be computed only once at the end of the procedure.

3.2 $\ell_{p,\epsilon}$-subspace estimation

The results for subspace estimation will be derived via a quadratic lifting of the classification problem of assigning point indexes to $I_1(\boldsymbol{B})$.

Definition 1 (Veronese map of degree 2)**.** *The Veronese map of degree 2 is the map*

$$\begin{aligned}\nu : \mathbb{R}^d &\to \mathbb{R}^D,\\ \boldsymbol{x} &\mapsto [x_1^2, x_1x_2, \ldots,\ x_2^2, x_2x_3, \ldots,\ x_{d-1}^2, x_{d-1}x_d,\ x_d^2]^T,\end{aligned}$$

where $D = d(d+1)/2$.

Using this map, we build a new data set

$$\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^N, \quad \text{with } \boldsymbol{z}_i = \begin{bmatrix} -\epsilon^2 \\ \nu(\boldsymbol{x}_i)\end{bmatrix} \tag{16}$$

and establish a correspondence between subspace estimation and the classification of this data set.

Lemma 3. *Given a subspace basis* $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d_s}] \in \mathbb{R}^{d\times d_S}$ *such that* $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}$ *and a set* $\mathcal{Z}$ *as in* (16)*, the set* $I_1(\boldsymbol{B})$ *defined in* (7) *is given by*

$$I_1(\boldsymbol{B}) = \{i \in I : \boldsymbol{h}(\boldsymbol{B})^T\boldsymbol{z}_i < 0\}, \tag{17}$$

where $h_1(\boldsymbol{B}) = 1$ *and* $\boldsymbol{h}_{2:D+1}(\boldsymbol{B}) = \boldsymbol{s} - \sum_{j=1}^{d_s}\nu(\boldsymbol{b}_j)$ *with the selection vector*[4] $\boldsymbol{s} \in \{0,1\}^D$ *such that* $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\nu(\boldsymbol{x})^T\boldsymbol{s} = \sum_{k=1}^d x_d^2$.

Proof. For any $i \in I$, we have

$$\begin{aligned}\|(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i\| < \epsilon &\Leftrightarrow \boldsymbol{x}_i^T(I - \boldsymbol{B}\boldsymbol{B}^T)^T(I - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i < \epsilon^2\\ &\Leftrightarrow \boldsymbol{x}_i^T(I - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{x}_i < \epsilon^2\\ &\Leftrightarrow \boldsymbol{x}_i^T\boldsymbol{x}_i - \sum_{j=1}^{d_s}\boldsymbol{x}_i^T(\boldsymbol{b}_j\boldsymbol{b}_j^T)\boldsymbol{x}_i < \epsilon^2\\ &\Leftrightarrow \nu(\boldsymbol{x}_i)^T\boldsymbol{s} - \sum_{j=1}^{d_s}\nu(\boldsymbol{x}_i)^T\nu(\boldsymbol{b}_j) < \epsilon^2\\ &\Leftrightarrow \nu(\boldsymbol{x}_i)^T\left[\boldsymbol{s} - \sum_{j=1}^{d_s}\nu(\boldsymbol{b}_j)\right] - \epsilon^2 < 0\\ &\Leftrightarrow \boldsymbol{h}(\boldsymbol{B})^T\boldsymbol{z}_i < 0.\end{aligned}$$

Thus, the definition of $I_1(\boldsymbol{B})$ coincides with (17). □

We will also need the following manipulation of Assumption 2.

Lemma 4. *Under Assumption 2, no hyperplane of* $\mathbb{R}^{D+1}$ *passes through the origin and more than* D *points of* $\mathcal{Z}$ *as defined in* (16).

Proof. Assumption 2 and the fact that the Veronese map is biregular, i.e., the image of points in general position in $\mathbb{R}^d$ under the Veronese map ν are again in general position, imply that $\mathcal{V} = \{\nu(\boldsymbol{x}_i)\}_{i=1}^N$ is in general position in $\mathbb{R}^D$. Therefore, there is no hyperplane of $\mathbb{R}^D$ that passes through more than D points of this set. Since the projection onto $\mathbb{R}^D$ of any hyperplane of $\mathbb{R}^{D+1}$ passing through the origin and more than D points of $\mathcal{Z}$ must pass through more than D points of $\mathcal{V}$, there is no such hyperplane. □

Proposition 3. *Under Assumption 2, in* $\mathbb{R}^{D+1}$*, there is a hyperplane of normal* $\boldsymbol{h} \in \mathbb{R}^{D+1}$ *passing through the origin and exactly* D *points of* $\mathcal{Z}$ *as in* (16) *such that a global solution to Problem 2 can be found by solving* 2^D *subspace estimation subproblems* (10).

[4] $\boldsymbol{s}$ is defined by $s_l = 1$ iff $l \in \{l_k\}_{k=1}^D$ with $l_1 = 1$ and $l_k = l_{k-1} + d - k + 2$.

Proof sketch. The proof works as the first part of the one of Proposition 2, except that we use Lemma 3 instead of Lemma 1 and Lemma 4 instead of Lemma 2 to bound from above the number of points lying on the hyperplane by D instead of $2d$. It is also simpler as the classification of $\boldsymbol{x}_i$ is directly given by the one of $\boldsymbol{z}_i$ without taking into account an additional $\boldsymbol{z}_{i+N}$. □

Compared with the regression case and Proposition 2, Proposition 3 does not ensure $h_1 > 0$ and thus the algorithm needs to test both orientations for every hyperplane. This yields Algorithm 2 for which we have the following result.

Theorem 2. *Under Assumption 2 and given that subproblem (10) can be solved in $\mathcal{O}(N^c)$ time for a constant $c \geq 1$ independent of N, Algorithm 2 solves Problem 2 in $\mathcal{O}(N^{c+d(d+1)/2})$ operations.*

Proof. For all subsets of D points of $\mathcal{Z}$, Algorithm 2 computes the hyperplane passing through the points. By Proposition 3, at least one of these hyperplanes with normal $\boldsymbol{h}$ is such that a global minimizer $\hat{\boldsymbol{B}}$ can be recovered by solving an instance of (10) in the inner loops over S and s.

The computational complexity, $T(N)$, of Algorithm 2 is the number of subsets of D points among N, $\binom{N}{D}$, times the time needed for computing a normal vector $\boldsymbol{h}$, $\mathcal{O}(D^3)$, classifying $\mathcal{Z}$ with $\boldsymbol{h}$, $\mathcal{O}(DN)$, and performing the inner loops over S and s with the subproblem (10), $\mathcal{O}(2 \times 2^D N^c)$:

$$T(N) = \mathcal{O}\left(\binom{N}{D}(D^3 + DN + 2^{D+1}N^c)\right) = \mathcal{O}(N^{D+c})$$

Recalling that $D = d(d+1)/2$ completes the proof. □

As an example, applying Theorem 2 for the saturated square loss, $\ell_{2,\epsilon}$, yields the exact Algorithm 2 for robust subspace estimation with complexity in the order of $\mathcal{O}(N^{1+d(d+1)/2})$.

Algorithm 2 Exact $\ell_{p,\epsilon}$-subspace estimation

Input: a data set $\{\boldsymbol{x}_i\}_{i=1}^N$, a threshold $\epsilon > 0$.
  Initialize $J^* \leftarrow \epsilon^2 N$.
  for all $S_h \subset \mathcal{Z}$ with $|S_h| = D$ **do**
    Compute the normal $\boldsymbol{h}$ of the hyperplane passing through the points in $S_h \cup \{\boldsymbol{0}\}$.
    Classify the points: $\forall i \in I$, $q_i = \text{sign}_0(\boldsymbol{h}^T \boldsymbol{z}_i)$.
    for all $S \subseteq S_h$ **do**
      for all $s \in \{-1, +1\}$ **do** *# for all orientation*
        $I_1 = \{i \leq N : q_i = s\} \cup \{i \leq N : \boldsymbol{x}_i \in S\}$
        Compute $\hat{\boldsymbol{B}}$ as in (10) with I_1 replacing $I_1(\boldsymbol{B}^*)$.
        if $J_p^S(\hat{\boldsymbol{B}}) < J^*$ **then**
          Update $J^* \leftarrow J_p^S(\hat{\boldsymbol{B}})$, $\boldsymbol{B}^* \leftarrow \hat{\boldsymbol{B}}$.
        end if
      end for
    end for
  end for
  return $J^*, \boldsymbol{B}^*$

4 Random sampling

We now detail more practical (but approximate) variants of the algorithms above. These are based on random sampling of subsets of points rather than a complete enumeration. As such, they share some features with the well-known RANSAC method (Fischler and Bolles, 1981) for robust estimation.

RANSAC The RANSAC method iterates through small subsets of s points and estimates a linear model at each iteration from these s points only. Then, the model that best approximates the maximum number of points can be retained, the rationale being that it should be possible to find a small subset of points within the set inliers and thus to estimate a good model from inliers only. However, this approach has two major drawbacks. First, since only s points are used to estimate the models, even if all the $\binom{N}{s}$ subsets are completely enumerated, the RANSAC cannot guarantee the recovery of an optimal solution, unless s equals the *unknown* number of inliers. And in this case, the computational complexity becomes exponential in N as soon as we assume that a certain fraction of the N points are inliers. The second weakness is related to the tuning of s which should be made in accordance with the noise level and the fraction of inliers: larger values of s tend to filter more efficiently the noise but also generate more subsets corrupted by outliers and decrease the probability of selecting a subset of inliers only.

Random sampling variants of Algorithms 1–2 Inspired by the RANSAC method, we can develop approximate variants of Algorithms 1–2, in which the complete enumeration of the subsets S is replaced by random sampling. Note that though this approach also relies on randomly sampling subsets of points, it remains quite different from the RANSAC. Indeed, the RANSAC directly estimates a linear model from the points in a subset, whereas here the subsets of points are only used to determine the classification of the entire data set into inliers and outliers. Therefore, it remains possible for such approximate variants to find the optimal set of inliers from which it can compute an exact solution. As in the RANSAC, the accuracy of the resulting model mostly depends on the number N_{iters} of iterations and thus of tested subsets. However, contrarily to the RANSAC, there is no other hyperparameter to tune: the size of the subsets is fixed to d (for regression) or D (for subspace estimation) from the analysis of Sect. 3.

5 Experiments

This section reports a few numerical results regarding first the exact algorithm for regression and then its approximate variant based on random sampling.

5.1 Exact algorithm

We here evaluate the gain in computing time offered by the exact polynomial algorithms. In particular, we focus on Algorithm 1 for $\ell_{0,\epsilon}$-regression and compare its computing time with that needed to solve Problem 1 with standard tools that can also compute exact solutions such as mixed-integer programming solvers. Specifically, for $p = 0$, Problem 1 can be reformulated as the mixed-integer linear program (MILP)

$$\begin{aligned} \min_{\boldsymbol{w}\in[-W,W]^d,\boldsymbol{\beta}\in\{0,1\}^N} & \sum_{i=1}^{N} \beta_i \\ s.t.\ & y_i - \boldsymbol{w}^T\boldsymbol{x}_i - \epsilon \leq M\beta_i, \quad i = 1,\dots,N, \\ & \boldsymbol{w}^T\boldsymbol{x}_i - y_i - \epsilon \leq M\beta_i, \quad i = 1,\dots,N, \end{aligned} \tag{18}$$

with binary variables β_i encoding $\mathbf{1}_{|y_i-\boldsymbol{w}^T\boldsymbol{x}_i|<\epsilon}$ and a constant M large enough to upper bound any absolute error term, i.e., set as $M \geq \max_{i\in\{1,\dots,N\}} |y_i| + dW\|\boldsymbol{x}_i\|_\infty$.

Experiments are conducted for random data sets of increasing size N generated by $y_i = \boldsymbol{x}_i^T\boldsymbol{w}_0 + \xi_i + \nu_i$ for $\boldsymbol{x}_i$ uniformly distributed in $[-5,5]^d$, $\boldsymbol{w}_0 = [1,-0.5,0.8]^T$, a zero-mean Gaussian noise ξ_i of variance 0.1 and an outlying Gaussian noise ν_i of mean 100 and variance 1000 added only to 40% of the data. Computing times reported in Table 1 refer to a parallel implementation in Matlab of Algorithm 1 and the use of CPLEX for solving (18) (which also benefits from parallel processing) on a laptop equipped with an i5-7440HQ processor at 2.8GHz.

While CPLEX can compete with Algorithm 1 on small data sets ($N = 100$), the worst-case exponential complexity of MILPs makes it far slower when N increases. In addition, its computing time highly varies between different trials for the same data set size, whereas that of Algorithm 1 is not influenced by the data and remains pretictable for given problem dimensions.

Table 1: Comparison of the computing time (mean ± standard deviation over 4 trials) of Algorithm 1 and that of solving the MILP (18). "n/a" appears when the method did not terminate after 10 hours.

N	100	300	1000
MILP (18)	0.5 ± 0.2 s	11.3 ± 18.5 min	n/a
Algorithm 1	3.9 ± 0.1 s	43.2 ± 0.8 s	25 ± 0.4 min

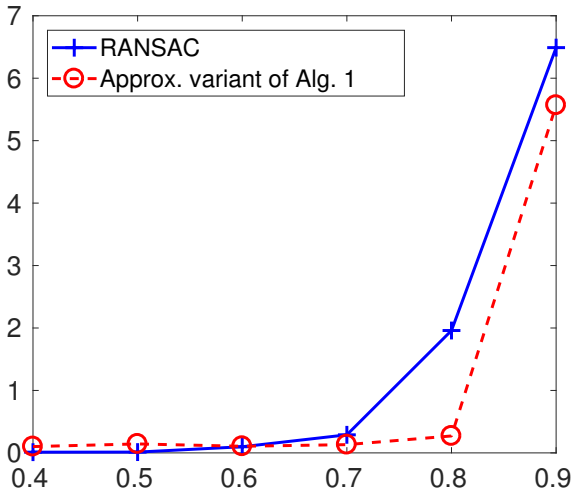

Figure 1: Error versus the fraction of outliers.

5.2 Approximate variant

We now compare the variant of Algorithm 1 for $p = 2$ proposed in Sect. 4 with the RANSAC when both algorithms use the same number of iterations ($N_{iters} = 3000$). Figure 1 shows how the errors, $\|\boldsymbol{w}_0 - \boldsymbol{w}\|_2/\|\boldsymbol{w}_0\|_2$, between the target vector $\boldsymbol{w}_0$ and the estimate $\boldsymbol{w}$ evolve with the fraction r of outliers ($\nu_i \neq 0$ for rN data points). More precisely, we plot the mean errors over 100 trials with $\boldsymbol{w}_0$ uniformly drawn in $[-5, 5]^d$, $d = 4$. The RANSAC uses $s = 2d$, but similar results are obtained for other values. The plot in Fig. 1 indicates that the proposed random sampling can perform better than the RANSAC in highly perturbed regimes with 70% of outliers or more. In addition, the random sampling variant of Algorithm 1 is obviously much faster than its exact version, leading to computing times similar to those of RANSAC and about 0.1 second in these experiments.

6 Conclusions

The paper analyzed the complexity of globally minimizing saturated loss functions for robust regression and subspace estimation with respect to the number of data. By deriving explicit connections between these estimation problems and linear classification, the paper could build on recent results on the enumeration of linear classifications to show that these global optimization problems have no more than a polynomial complexity in the number of data. Experiments showed that this provides a significant gain in speed when compared to a mixed-integer programming approach.

However, the developed algorithms have an exponential complexity wrt. the data dimension, which strongly limits their practical use. Therefore, approximate variants were proposed based on random sampling. Experiments showed that these variants can yield an increase of accuracy for a similar computational cost when compared with another classical method based on random sampling, namely, the RANSAC.

References

Bako, L., 2011. Identification of switched linear systems via sparse optimization. Automatica 47, 668–677.

Bako, L., 2014. Subspace clustering through parametric representation and sparse optimization. IEEE Transactions on Signal Processing 21, 356–360.

Bako, L., Ohlsson, H., 2016. Analysis of a nonsmooth optimization approach to robust estimation. Automatica 66, 132–145.

Bemporad, A., Garulli, A., Paoletti, S., Vicino, A., 2005. A bounded-error approach to piecewise affine system identification. IEEE Transactions on Automatic Control 50, 1567–1580.

Bhatia, K., Jain, P., Kar, P., 2015. Robust regression via hard thresholding, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 28, pp. 721–729.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381395.

Hössjer, O., 1995. Exact computation of the least trimmed squares estimate in simple linear regression. Computational Statistics & Data Analysis 19, 265–282.

Lauer, F., 2015. On the complexity of piecewise affine system identification. Automatica 62, 148–153.

Lauer, F., 2016. On the complexity of switching linear regression. Automatica 74, 80–83.

Lauer, F., Bloch, G., 2018. Hybrid System Identification: Theory and Algorithms for Learning Switching Models. Springer.

Li, L., 2005. An algorithm for computing exact least-trimmed squares estimate of simple linear regression with constraints. Computational Statistics & Data Analysis 48, 717–734.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y., 2013. Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 171–184.

Rousseeuw, P.J., Leroy, A.M., 2005. Robust regression and outlier detection. John Wiley & Sons.

Shevlyakov, G., Morgenthaler, S., Shurygin, A., 2008. Redescending M-estimators. Journal of Statistical Planning and Inference 138, 2906–2917.

Vidal, R., 2011. Subspace clustering. IEEE Signal Processing Magazine 28, 52–68.