

Minimum Sum of Distances Estimator: Robustness and Stability

Yoav Sharon, John Wright, Yi Ma

Abstract—We consider the problem of estimating a state x from noisy and corrupted linear measurements $\mathbf{y} = A\mathbf{x} + \mathbf{z} + \mathbf{e}$, where \mathbf{z} is a dense vector of small-magnitude noise and \mathbf{e} is a relatively sparse vector whose entries can be arbitrarily large. We study the behavior of the ℓ^1 estimator $\hat{\mathbf{x}} = \arg \min_x \|\mathbf{y} - A\mathbf{x}\|_1$, and analyze its breakdown point with respect to the number of corrupted measurements $\|\mathbf{e}\|_0$. We show that under mild conditions, the breakdown point does not depend on the noise level. We introduce a novel algorithm for computing the breakdown point for any given A , and provide a simple bound on the estimation error when the number of corrupted measurements is less than the breakdown point. We apply our algorithm to design a robust state estimator for an autonomous vehicles, and show how it can significantly improve performance over the Kalman filter.

I. INTRODUCTION

The problem of estimating a state $\mathbf{x}_0 \in \mathbb{R}^n$ from $m > n$ noisy linear measurements $\mathbf{y} \approx A\mathbf{x}_0 \in \mathbb{R}^m$, arises in a vast number of applications. In some applications one can assume that the difference between \mathbf{y} and $A\mathbf{x}_0$ is a small i.i.d. Gaussian noise $\mathbf{z} \in \mathbb{R}^m$:

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{z}. \quad (1)$$

In this case, the optimal estimate of \mathbf{x}_0 is the least-squares estimate: $\hat{\mathbf{x}}_2 = (A^T A)^{-1} A^T \mathbf{y} = \arg \min_x \|\mathbf{y} - A\mathbf{x}\|_2$. The least-square estimate is known as *stable* in the sense that the estimation error $\|\hat{\mathbf{x}}_2 - \mathbf{x}_0\|_2$ is bounded by a continuous function in \mathbf{z} . Thus, small noise causes only small estimation error. Often, however, some of the measurements in \mathbf{y} can be corrupted by arbitrarily large errors. In this case, we instead must solve \mathbf{x}_0 from the equation

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{z} + \mathbf{e}, \quad (2)$$

where $\mathbf{e} \in \mathbb{R}^m$ has some arbitrarily large nonzero entries. One typical example is a GPS system, whose estimated position output can occasionally be considerably corrupted when the signals from the satellites are reflected off the surrounding terrain (i.e. multipath). Even one such corrupted measurement can cause arbitrarily large estimation error in the least-squares estimate.

When the state being estimated is a scalar ($n = 1$), the least-squares estimate \hat{x}_2 is equivalent to taking a weighted average of the measurements. A known robust alternative to the average is the median. With the median, up to almost 50% of the measurements can be arbitrarily corrupted before the

estimation error becomes unbounded. That is, the breakdown point of the median is 50%.

Taking the median, one essentially looks for the point which minimizes the sum of distances to all the measurements whereas taking the average minimizes sum of the squares of these distances. One natural generalization of this concept to multivariate ($n > 1$) estimation¹ is to view the m measurements $\mathbf{y} \doteq [y_1, \dots, y_m]^T$ as defining m hyperplanes:

$$H_i \doteq \{\mathbf{x} \in \mathbb{R}^n \mid y_i = \mathbf{a}_i^T \mathbf{x}\},$$

where $\mathbf{a}_i^T \in \mathbb{R}^n$ is the corresponding row of the matrix $A \doteq [\mathbf{a}_1, \dots, \mathbf{a}_m]^T$. Then the “median” estimate for \mathbf{x} can be defined to be the point that minimizes the sum of distances to these hyperplanes:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \sum_{i=1}^m |y_i - \mathbf{a}_i^T \mathbf{x}| = \arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_1. \quad (3)$$

To understand why this estimate can be robust to errors, let us assume the noise is zero for now: $\mathbf{z} = \mathbf{0}$. That is, we try to solve \mathbf{x}_0 from the equation $\mathbf{y} = A\mathbf{x}_0 + \mathbf{e}$. If we could somehow compute \mathbf{e} , then \mathbf{x}_0 could be easily recovered from the clean system of equations $A\mathbf{x}_0 = \mathbf{y} - \mathbf{e}$. One approach to recovering \mathbf{e} is to choose a matrix $B \in \mathbb{R}^{p \times m}$, $p = m - n$, with $BA = 0$, and define $\mathbf{w} = B\mathbf{y}$. Multiplying both sides of the measurement equation by B yields an underdetermined system of equations $\mathbf{w} = B\mathbf{e}$ in \mathbf{e} alone. In the context of compressed sensing [2], it has recently been discovered that whenever \mathbf{e} is sparse enough, it can be correctly recovered by solving the following ℓ^1 -minimization problem:

$$\hat{\mathbf{e}} = \arg \min_{\mathbf{e}} \|\mathbf{e}\|_1 \quad \text{subject to } \mathbf{w} = B\mathbf{e}. \quad (4)$$

So, in the noise free case, the two problems (3) and (4) are equivalent.

There is also a large literature analyzing the performance of (4) and related estimates in the presence of noise. The strongest available results ([3], [4], amongst others) have the following flavor: for some constants C and ρ , and almost all random matrices B , if one applies an ℓ^2 -penalized version of (4) (i.e., the Lasso [5], [6]) and the number of errors $\|\mathbf{e}\|_0$ is less than $\rho \cdot n$, then the estimation error is bounded by $C \cdot \|\mathbf{z}\|$ for some $C > 0$. However, specific forms of the constants C and ρ are difficult to derive. A similar bound

¹Another multivariate generalization of the median occurs in robust centerpoint estimation, where the observations are themselves points (rather than inner products). There, the estimator that minimizes the sum of distances to the observations, known as the Fermat-Weber point, achieves a breakdown point of 50% [1, Theorem 2.2]. Although the estimator studied here also generalizes the median, it addresses the more general problem of robust linear regression.

can be derived when B is known to be a *restricted isometry* [3]. However, it requires prior knowledge of the noise level, and the estimation error depends on the number of corrupted measurements, with the bound C diverging to infinity when the error fraction ρ approaches the breakdown point. Similar results have also been obtained for greedy alternatives to ℓ^1 -minimization [7]. In this setting, one does not require a bound on the noise term. However, it does require that the number of corrupted measurements be considerably lower than the breakdown point for ℓ^1 -minimization.

Whereas most of the existing stability results and bounds are derived for the underdetermined case (4), in this paper, we directly study the stability of the ℓ^1 estimator for the overdetermined problem (3). Our bounds are weaker than those obtained in the asymptotic setting of large random matrices and small error fractions [4]. However, they hold for all matrices A , including the structured matrices arising in state estimation problems, and all error fraction ρ , up to the intrinsic breakdown point of the ℓ^1 estimator. Moreover, our bound has a very simple expression, whose derivation naturally suggests an algorithm for computing the intrinsic breakdown point of the ℓ^1 estimation. The complexity of our algorithm is exponentially faster than the existing alternative, and it is especially suitable for the kind of problems of interest for the system and control community – moderate-sized robust state estimation problems.

II. PRELIMINARIES

Throughout, the 0-norm will denote the number of nonzero elements in a vector $\mathbf{v} \in \mathbb{R}^m$:

$$\|\mathbf{v}\|_0 \doteq \#\{i | v_i \neq 0\}$$

We will use $[m]$ to denote the set of indices $[m] \doteq \{1, 2, \dots, m\}$. We will use the following notation for “positive” directional derivative of an arbitrary multivariate function $f: \mathbb{R}^m \rightarrow \mathbb{R}$:

$$D_{\mathbf{v}}^+ f(\mathbf{x}) = \lim_{\varepsilon \searrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon}.$$

Consider a general estimation problem, $\mathbf{y} = f(\mathbf{x}_0, \mathbf{z}, \mathbf{e})$, where \mathbf{x}_0 is the unknown state to be estimated, \mathbf{z} is a noise term, \mathbf{e} is a corruption term and \mathbf{y} is the available measurements. Let $\hat{\mathbf{x}} = g(\mathbf{y})$ be some estimate. We say that for given \mathbf{x}_0 and \mathbf{z} , the estimate is robust up to T corrupted measurements (or T -robust) if there exists a smooth function $\beta(\mathbf{x}_0, \mathbf{z}) \in \mathbb{R}$ such that:

$$\forall \mathbf{e}: \text{if } \|\mathbf{e}\|_0 < T \text{ then } \|\hat{\mathbf{x}} - \mathbf{x}_0\|_2 \leq \beta(\mathbf{x}_0, \mathbf{z}). \quad (5)$$

The *breakdown point* of this scheme, $T^*(\mathbf{x}_0, \mathbf{z})$, is the minimum $T \in \mathbb{N}$ for which the estimation scheme is *not* T -robust. In other words,

$$T^*(\mathbf{x}_0, \mathbf{z}) \doteq \min \left\{ T \in \mathbb{N} \mid \sup_{\mathbf{e}, \|\mathbf{e}\|_0 \leq T} \|\hat{\mathbf{x}} - \mathbf{x}_0\|_2 = \infty \right\}.$$

We say T^* is a *stable* breakdown point if it does not depend on \mathbf{x}_0 and \mathbf{z} , i.e. $T^*(\mathbf{x}_0, \mathbf{z}) \equiv T^*$.

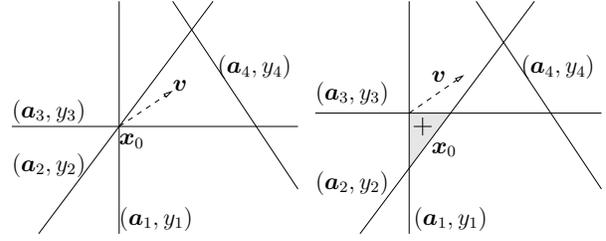


Fig. 1. Low-dimensional ($n = 2$) illustration of the main ideas: here, the 1st, 2nd and 3rd lines are uncorrupted measurement hyperplanes, while the 4th has been corrupted. In the **left** diagram there is no noise. The cost function will not be minimized at \mathbf{x}_0 if there exists a direction \mathbf{v} which increases the sum of distances to the uncorrupted hyperplanes at a rate slower the rate at which it decreases the sum of distances to the corrupted hyperplane(s). This condition is formulated in (8). In the **right** diagram there is also noise. The polytope P_T defined by the uncorrupted hyperplanes is shaded. The vector \mathbf{v} illustrates a possible direction which reduces the cost function, even if (11) holds. This is because going in this direction will also reduce the distance to 2nd hyperplane, which is uncorrupted. Note that the \mathbf{a}_i 's are vectors perpendicular to the hyperplanes they represent.

Throughout this paper we consider the problem of estimating \mathbf{x}_0 from \mathbf{y} :

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{z} + \mathbf{e},$$

where $\mathbf{x}_0 \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{e} \in \mathbb{R}^m$. For this problem we consider the Minimum Sum of Distances (MSoD) estimation scheme

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} C_{\mathbf{y}}(\mathbf{x}), \quad (6)$$

with cost function

$$C_{\mathbf{y}}(\mathbf{x}) \doteq \|\mathbf{y} - A\mathbf{x}\|_1. \quad (7)$$

Our goal is to study whether the breakdown point of this estimate is stable and if so, how to compute it.

We start by giving results pertaining to the noiseless case, $\mathbf{z} = \mathbf{0}$. We assume T of the measurements can be corrupted. Geometrically, this means that the remaining $m - T$ measurement hyperplanes $H_i \doteq \{\mathbf{x} \in \mathbb{R}^n \mid y_i = \mathbf{a}_i^T \mathbf{x}\}$ pass through \mathbf{x}_0 . We will let I denote the indices of these uncorrupted hyperplanes. The corrupted ones will be conveniently denoted by I^c . We ask whether these T hyperplanes can be positioned so that \mathbf{x}_0 no longer minimizes the cost function (7). Since $C_{\mathbf{y}}$ is convex, this will be true if and only there exists a direction \mathbf{v} , from \mathbf{x}_0 , along which the cost function decreases, i.e. $D_{\mathbf{v}}^+ C_{\mathbf{y}}(\mathbf{x}_0) < 0$. Since the uncorrupted hyperplanes pass through \mathbf{x}_0 , moving in the direction of \mathbf{v} from \mathbf{x}_0 will increase the distance to each of the uncorrupted hyperplanes at a rate of $|\mathbf{a}_i^T \mathbf{v}|$, $i \in I$. We have freedom in placing the corrupted hyperplanes, and so for each \mathbf{v} we can position them so that moving in the direction of \mathbf{v} will decrease the distance to each of the corrupted hyperplanes by a rate of $|\mathbf{a}_i^T \mathbf{v}|$, $i \in I^c$. In this case, which can be referred to as worst positioning of the corrupted hyperplanes given \mathbf{v} , the condition $D_{\mathbf{v}}^+ C_{\mathbf{y}}(\mathbf{x}_0) < 0$ becomes

$$\sum_{i \in I} |\mathbf{a}_i^T \mathbf{v}| - \sum_{i \in I^c} |\mathbf{a}_i^T \mathbf{v}| < 0. \quad (8)$$

This is illustrated by the left diagram in Figure 1. Because (8) represents the worst case for a given \mathbf{v} , \mathbf{x}_0 fails to minimize the cost function if and only if (8) holds for some \mathbf{v} . Thus we arrive at a lemma following the next definition:

Definition 2.1: $\tilde{T}(A)$ is defined as the minimal integer T for which there exists $I \subset [m]$, $|I| = m - T$ and $\mathbf{v} \in \mathbb{R}^n$ such that (8) holds.

Lemma 2.1: Under the condition $\mathbf{z} = \mathbf{0}$, the breakdown point of the estimation scheme (6) is equal to \tilde{T} as defined in Definition 2.1, i.e. $T^*(\mathbf{x}_0, \mathbf{0}) = \tilde{T}(A)$, $\forall \mathbf{x}_0 \in \mathbb{R}^n$.

In the next section we will consider the noisy case and show that this breakdown point is stable and the estimation error is bounded by a simple function $\beta(\mathbf{z})$ that does not depend on \mathbf{x}_0 .

III. PROOF OF ROBUSTNESS

Throughout this section, we will make the following assumptions.

Assumption 3.1: The hyperplanes are in general position: every set of n normals to the hyperplanes (the \mathbf{a}_i 's) spans \mathbb{R}^n . In other words, any n rows of A are linearly independent and form a nonsingular submatrix of A .

Matrices that violate this assumption are of zero measure. In practice, if a given matrix A has n linearly dependent rows, one can always remove a row (by viewing it as a redundant measurement).² In this way, we can always ensure the matrix A satisfies the above assumption.

Assumption 3.2: Unless the noise term is zero ($\mathbf{z} = \mathbf{0}$), no $n + 1$ hyperplanes passes through the same point – there does not exist a set $J \subset [m]$, $|J| = n + 1$ and a point $\mathbf{x} \in \mathbb{R}^n$ such that $y_i = \mathbf{a}_i^T \mathbf{x} \forall i \in J$.

Note that the set of \mathbf{z} 's that do not satisfy this assumption is of zero measure. The main result is as follows:

Theorem 3.1: Under Assumptions 3.1 and 3.2 the estimation scheme (6) has a stable breakdown point $T^* = \tilde{T}(A)$. Furthermore, $\beta(\mathbf{x}_0, \mathbf{z})$ in the definition of T -robustness has the form $\beta(\mathbf{x}_0, \mathbf{z}) = \alpha(A) \|\mathbf{z}\|_2$ where

$$\alpha(A) \doteq \max_{J \subset [m], |J|=n} \|A_J^{-1}\|. \quad (9)$$

This section is devoted to proving Theorem 3.1 through several steps of generalizations. In the next section we will present algorithms to calculate the breakdown point.

In the presence of noise ($\mathbf{z} \neq \mathbf{0}$) the uncorrupted hyperplanes no longer all pass through \mathbf{x}_0 . Instead, they create a polytope P_I around or near \mathbf{x}_0 . This polytope, P_I , is defined as the convex hull ($P_I \doteq \text{conv } Q_I$) of all the intersections of n uncorrupted hyperplanes:

$$Q_I \doteq \{ \mathbf{q}_J \in \mathbb{R}^n \mid \mathbf{q}_J = A_J^{-1} \mathbf{y}_J, J \subset I, |J| = n \}, \quad (10)$$

where $A_J \doteq [\mathbf{a}_{J_1}, \dots, \mathbf{a}_{J_n}]^T$ and $\mathbf{y}_J \doteq [y_{J_1}, \dots, y_{J_n}]^T$. See the right diagram in Figure 1 for an illustration.

Note that if $|I| > m - \tilde{T}(A)$ then

$$\forall \mathbf{v} \in \mathbb{R}^n : \sum_{i \in I} |\mathbf{a}_i^T \mathbf{v}| - \sum_{i \in I^c} |\mathbf{a}_i^T \mathbf{v}| \geq 0. \quad (11)$$

²However, it might be possible to revise our results so that they do not require such a preprocessing.

We will show that if (11) holds then $\hat{\mathbf{x}} \in P_I$.

Let $\mathbf{q}_J = A_J^{-1} \mathbf{y}_J$, $J \subset I$, $|J| = n$ be one of the intersections that define P_I . Because not all of the uncorrupted, noisy hyperplanes pass through \mathbf{q}_J , even if (11) does hold, it is possible that there exists a direction \mathbf{v} from \mathbf{q}_J that decreases the cost function, since going this direction might move the estimate \mathbf{x} closer to one of the uncorrupted but noisy hyperplanes (see Figure 1 right). First, we will show that if such a direction exists, then there exists a direction which decreases the cost function, but also points straight to another intersection (of n uncorrupted hyperplanes). This is stated in the following Lemma:

Lemma 3.1: Assume condition (11) holds for $I \subset [m]$. Let $\mathbf{q}_{J_0} = A_{J_0}^{-1} \mathbf{y}_{J_0}$ for some $J_0 \subset I$, $|J_0| = n$. If there exists a direction \mathbf{v} from \mathbf{q}_{J_0} which reduces the cost function, i.e. $D_{\mathbf{v}}^+ C_{\mathbf{y}}(\mathbf{q}_{J_0}) < 0$, then there also exists a direction \mathbf{v}' , and indices $j \in J_0$ and $k \in I \setminus J_0$, with the following properties:

- 1) $D_{\mathbf{v}'}^+ C_{\mathbf{y}}(\mathbf{q}_{J_0}) < 0$,
- 2) $\mathbf{a}_i^T \mathbf{v}' = 0 \forall i \in J_0 \setminus \{j\}$,
- 3) $\text{sgn}(\mathbf{a}_k^T \mathbf{v}') = \text{sgn}(y_k - \mathbf{a}_k^T \mathbf{v}')$.

We will leave the proof of this lemma to the Appendix and here we only discuss its implications. The first property says that \mathbf{v}' is a direction which also reduces the cost function. The second property says that if we start moving in direction \mathbf{v}' we will remain on all but one hyperplane from the set J_0 . The last property says that the direction \mathbf{v}' points to another uncorrupted hyperplane. Together, the last two properties guarantee that moving in direction \mathbf{v}' from \mathbf{q}_{J_0} , we will eventually reach another intersection of n uncorrupted hyperplanes. To better understand the last property, look at the right diagram in Figure 1. Assume \mathbf{q}_{J_0} is on the intersection of the 1st and 3rd hyperplanes ($J_0 = \{1, 3\}$), and $j = 1$. In this case the second property constrains us to move left or right along the 3rd hyperplane. Moving left, however, will take us away from the polytope P_I . The last property guarantees that we will move right, toward the 2nd hyperplane.

We start the proof of Theorem 3.1 by first assuming that none of the corrupted hyperplanes intersects with the polytope P_I :

Proposition 3.1: Assume condition (11) holds for $I \subset [m]$. If all the corrupted hyperplanes $H_i, i \in I^c$ do not intersect with the polytope P_I , then the minimizer $\hat{\mathbf{x}}$ is at an intersection of n uncorrupted hyperplanes.

Proof: Consider some intersection \mathbf{q}_{J_0} of n uncorrupted hyperplanes. If there is no direction \mathbf{v} away from \mathbf{q}_{J_0} along which the convex function $C_{\mathbf{y}}$ decreases, \mathbf{q}_{J_0} is the global minimizer. If, however, there is such a direction \mathbf{v} , then by Lemma 3.1 there is also a \mathbf{v}' such that the ray $\{\mathbf{q}_{J_0} + \lambda \mathbf{v}' \mid \lambda \geq 0\}$ lies in the intersection of $n - 1$ uncorrupted hyperplanes, and $\mathbf{q}' \doteq \mathbf{q}_{J_0} + \lambda^* \mathbf{v}' \in H_i$ for some $\lambda^* > 0$ and some uncorrupted hyperplane with $i \notin J_0$. Since \mathbf{q}' is the intersection of n uncorrupted hyperplanes, the segment $[\mathbf{q}_{J_0}, \mathbf{q}'] \subset P_I$. If we start at \mathbf{q}_{J_0} and move along this segment, $C_{\mathbf{y}}$ decreases linearly at a rate of $D_{\mathbf{v}'}^+(C_{\mathbf{y}})(\mathbf{q}_{J_0}) < 0$ until we reach another measurement hyperplane (possibly H_i). Since none of the corrupted hyperplanes intersects P_I ,

we have reached an intersection of n uncorrupted hyperplanes.

Repeat this process, at each step moving to an intersection \mathbf{q}_J with smaller $C_{\mathbf{y}}$. Because the number of such intersections is finite, the process terminates at some \mathbf{q}_J , from which there cannot exist a direction \mathbf{v} along which $C_{\mathbf{y}}$ decreases. This intersection is the global minimizer of $C_{\mathbf{y}}$. ■

From the above proposition, the only case when the minimizer $\hat{\mathbf{x}}$ can be outside of the polytope P_I is when at least one of the corrupted hyperplanes intersects P_I . However, the following lemma rules out that possibility.

Lemma 3.2: Assume condition (11) holds for $I \subset [m]$. If the minimizer $\hat{\mathbf{x}}$ is not in P_I , then it is possible to move all the corrupted hyperplanes (by modifying \mathbf{y}_{I^c}) so that they no longer intersect P_I , and furthermore, the new minimizer remains outside P_I .

The conclusion of this lemma obviously creates a situation that contradicts the previous proposition. Thus, its assumption must be invalid and we can conclude that the minimizer must always be inside the polytope P_I . We delay the proof of this lemma to the Appendix.

Taking the above lemmas for granted, we are now ready to put together the proof for the main theorem. In the proof below (and proofs of the lemmas in the appendix), we will use the following observation.

Remark 3.1: Without loss of generality, for a given $\mathbf{y} \in \mathbb{R}^m$ and arbitrary direction $\mathbf{v} \in \mathbb{R}^n$, we can assume that $\mathbf{a}_i^T \mathbf{v} \geq 0 \forall i \in [m]$. This is because we can arbitrarily negate some of the \mathbf{a}_i 's and their corresponding y_i 's without affecting the cost function (7).

Proof of Theorem 3.1: As discussed above, combining Proposition 3.1 and Lemma 3.2, we may conclude that if $\|e\|_0 < \tilde{T}(A)$ (so condition (11) holds), then $\hat{\mathbf{x}} \in P_I$. Therefore, we can bound the estimation error as:

$$\begin{aligned} \|\hat{\mathbf{x}} - \mathbf{x}_0\|_2 &\leq \max_{\mathbf{x} \in P_I} \|\mathbf{x} - \mathbf{x}_0\|_2 = \max_{J \subset I, |J|=n} \|A_J^{-1} \mathbf{y}_J - \mathbf{x}_0\|_2 \\ &= \max_{J \subset I, |J|=n} \|A_J^{-1} (A_J \mathbf{x}_0 + \mathbf{z}_J) - \mathbf{x}_0\|_2 \\ &= \max_{J \subset I, |J|=n} \|A_J^{-1} \mathbf{z}_J\|_2 \leq \max_{J \subset [m], |J|=n} \|A_J^{-1}\| \|\mathbf{z}_J\|_2, \end{aligned}$$

where $\|\cdot\|$ is the operator norm induced by the 2-norm. This proves that $T^*(\mathbf{x}_0, \mathbf{z}) \geq \tilde{T}$ for all \mathbf{x}_0 and \mathbf{z} . Furthermore, for $T = \tilde{T}$ we can use $\beta(\mathbf{x}_0, \mathbf{z})$ as defined in Theorem 3.1 to show that (5) holds.

We complete the proof by showing that $T^*(\mathbf{x}_0, \mathbf{z}) \leq \tilde{T}$ for all \mathbf{x}_0 and \mathbf{z} . By definition of \tilde{T} , there exists $I \subset \{1 \dots m\}$, $|I| = m - \tilde{T}$ and \mathbf{v} such that (8) holds. Let \mathbf{x}_0 , \mathbf{z} and τ be given. WLOG (Remark 3.1), assume $\mathbf{a}_i^T \mathbf{v} \geq 0 \forall i$. Set

$$e_i \doteq \begin{cases} -z_i + \tau, & i \notin I, \\ 0, & i \in I, \end{cases} \quad (12)$$

so that $\|e\|_0 = \tilde{T}$. Note that this choice of \mathbf{e} puts the corrupted hyperplanes (with indices I^c) at distance of τ from \mathbf{x}_0 . For every point $\tilde{\mathbf{x}}$ at (Euclidean) distance less than τ from \mathbf{x}_0 we have $\mathbf{a}_i^T \tilde{\mathbf{x}} < y_i$ and $\mathbf{a}_i^T \mathbf{v} \geq 0 \forall i \in I^c$ which means going at direction \mathbf{v} brings us closer to all the corrupted

hyperplanes. Thus,

$$D_{\mathbf{v}}^+ C_{\mathbf{y}}(\mathbf{x}_0) \leq \sum_{i \in I} |\mathbf{a}_i^T \mathbf{v}| - \sum_{i \in I^c} |\mathbf{a}_i^T \mathbf{v}| < 0,$$

so $\tilde{\mathbf{x}}$ can not minimize the cost function. Therefore we must have $\|\hat{\mathbf{x}} - \mathbf{x}\| \geq \tau$, and since τ can be made arbitrarily large, this proves that indeed $T^*(\mathbf{x}_0, \mathbf{z}) \leq \tilde{T}$. ■

Theorem 3.1 reveals an interesting ‘‘separation principle’’ for the sum of distances estimate $\hat{\mathbf{x}} = \arg \min \|\mathbf{y} - A\mathbf{x}\|_1$: For the model $\mathbf{y} = A\mathbf{x}_0 + \mathbf{z} + \mathbf{e}$, when it comes to robustness to corruption \mathbf{e} , one may evaluate its breakdown point as if the noise \mathbf{z} were zero; when it comes to stability to noise \mathbf{z} , one may bound the worst-case error in the estimate as if there was no error \mathbf{e} (as long as its support is below the breakdown point).

Thus, given a matrix A and the associated system of linear equations, the breakdown point $T^*(A)$ and the error bound $\alpha(A)$ fully characterize when and how the sum of distances estimate works. In the next section, we show how to compute or estimate these crucial quantities associated with A .

IV. COMPUTING THE BREAKDOWN POINT

Definition 2.1 does not immediately suggest an algorithm for computing $\tilde{T} = T^*$, because it requires checking condition (8) for all $\mathbf{v} \in \mathbb{R}^n$, $\|\mathbf{v}\|_2 = 1$, and there are infinite number of such \mathbf{v} 's. The following corollary of Lemma 3.1, however, states that it is sufficient to check only a finite subset of \mathbb{R}^n :

Corollary 4.1: Condition (8) holds for some $I \subset [m]$ and $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ if and only if it holds for some $\mathbf{v}' \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ for which there exists $J \subset [m]$, $|J| = n - 1$ such that $\mathbf{a}_i^T \mathbf{v}' = 0 \forall i \in J$.

Proof: The *if* direction is trivial. For the *only if* direction assume (8) holds for some $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. Assume, WLOG (Remark 3.1), $\mathbf{a}_i^T \mathbf{v} \geq 0 \forall i$. Now choose $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{z} = \mathbf{0}$ and \mathbf{e} according to (12) with arbitrary $\tau > 0$. Also choose arbitrarily $J_0 \subset I$, $|J_0| = n$. With these choices $D_{\mathbf{v}}^+ C_{\mathbf{y}}(\mathbf{0}) < 0$ and (8) become equivalent. Thus we can use Lemma 3.1 to derive the existence of \mathbf{v}' for which (8) holds (Property 1 of the lemma) and also $\mathbf{a}_i^T \mathbf{v}' = 0 \forall i \in J \doteq J_0 \setminus \{j\}$ (Property 2). ■

Given $J \subset I$, $|J| = n - 1$, the condition $A_J \mathbf{v}' = \mathbf{0}$ determines \mathbf{v}' uniquely up to scale. The validity of condition (8) is unchanged by scaling \mathbf{v}' . Thus, we could equivalently define $T^*(A)$ to be the maximal integer T such that for all $J \subset [m]$ of size $|J| = n - 1$ and all $I \subset [m]$ of size $|I| = m - T$, condition (8) does not hold for any \mathbf{v}' satisfying $A_J \mathbf{v}' = \mathbf{0}$. Fix J (and a corresponding \mathbf{v}), and sort the $|\mathbf{a}_i^T \mathbf{v}|$ such that $|\mathbf{a}_{r_1}^T \mathbf{v}| \geq |\mathbf{a}_{r_2}^T \mathbf{v}| \geq \dots \geq |\mathbf{a}_{r_m}^T \mathbf{v}|$. Then, condition (8) holds for some I of size $T - m$ if and only if it holds for $I \doteq \{r_{T+1} \dots r_m\}$. We can therefore compute $T^*(A)$ by checking this condition for every subset J of size $n - 1$. This idea is formalized as Algorithm (1) below:

Algorithm 1 Computing $T^*(A)$

Input: $A \in \mathbb{R}^{m \times n}$.

- 1: Set $T \leftarrow m$ and let J_1, \dots, J_N , $N = \binom{m}{n-1}$, be all the subsets of $[m] \doteq \{1 \dots m\}$ containing $n-1$ indices.
- 2: **for** $k = 1 : N$ **do**
- 3: Find a nontrivial solution $v \in \mathbb{R}^n$ such that $a_i^T v = 0 \forall i \in J_k$.
- 4: Find the order $r_1 \dots r_m$ such that $|a_{r_1}^T v| \geq |a_{r_2}^T v| \geq \dots \geq |a_{r_m}^T v|$.
- 5: Find the largest integer, s , such that
$$\sum_{i=1}^s |a_{r_i}^T v| < \sum_{i=s+1}^m |a_{r_i}^T v|.$$
- 6: Set $T \leftarrow \min\{T, s\}$.
- 7: **end for**

Output: T .

The computation time of Algorithm 1 is

$$\binom{m}{n} (t_{sle}(n-1) + t_{mv}(m) + t_{sort}(m)), \quad (13)$$

where $t_{sle}(n) = \mathcal{O}(n^3)$, $t_{mv}(n) = \mathcal{O}(n^2)$ and $t_{sort}(n) = \mathcal{O}(n \log n)$ are the times it takes to solve a system of linear equations, to compute a matrix-vector multiplication, and to sort, respectively. When both m and n grow, $\binom{m}{n}$, and thus the computation time of our algorithm, grows exponentially. In many control applications, however, the number of variables describing the state of the system, n , is fixed, while the number of measurements, m , is flexible. In this case, where n is fixed, our algorithm's computation time is polynomial in m . We further note, that while the running time of the algorithm might still be relatively large in practice, from the engineering design point of view it needs to be executed only once during the design of the system to analyze its performance. In real-time only (6) needs to be evaluated, which can be done very efficiently using linear programming.

The algorithm described above is different from the existing algorithm in the literature for computing the breakdown point. In the introduction we have mentioned that in the absence of noise, (3) and (4) are equivalent problems when $B \in \mathbb{R}^{p \times m}$, $p = m - n$, $BA = 0$. The following result, proved in [8] and in [2, §II], states that the ability of (4) to recover e from the underdetermined linear system $w = Be$ depends only on the sign pattern of e :

Theorem 4.1: If for some $e' \in \mathbb{R}^n$, we have

$$e' = \arg \min_e \|e\|_1 \quad \text{subject to} \quad Be = Be', \quad (14)$$

then for all \tilde{e} such that $\text{sign}(\tilde{e}_i) = \text{sign}(e'_i)$, $i = 1 \dots n$,

$$\tilde{e} = \arg \min_e \|e\|_1 \quad \text{subject to} \quad Be = B\tilde{e}.$$

From this result, to determine whether we can recover any T -sparse signal e (i.e. $\|e\|_0 = T$), we only need to check one e for each T -sparse sign pattern. Specifically:

$$T^* = \max \left\{ T \in \mathbb{N} \mid \forall e' \in E_T : e' = \arg \min_{e \mid Be = Be'} \|e\|_1 \right\} \quad (15)$$

where $E_T \doteq \{e \in \mathbb{R}^m \mid \forall i : e_i \in \{-1, 0, 1\}, \|e\|_0 = T\}$.

Since $|E_T| = 2^T \binom{m}{T}$, a straightforward algorithm for computing (15) requires time

$$\sum_{T=1}^{T^*} 2^T \binom{m}{T} t_{lp}(m \times p), \quad (16)$$

where t_{lp} is the time it takes to solve the linear programming problem (14). We note that instead of actually solving for the right hand side of (14), one can check if e' minimizes the right hand side by looking for appropriate sub-gradients (see [2, §II]). This alternative approach, however, still requires solving a linear programming problem of similar size.

It is easy to see that the running time of our algorithm (13) is exponentially faster than the alternative (16) when n/m is small compared to T^*/m (i.e. A is very tall) or when n/m is very close to one (i.e. A is almost square). The first case is precisely the interest of robust estimation – the number of measurements needs to be large so as to tolerate more errors. This is the case for the robust state estimation problem one often encounters in control systems.

V. COMPARISON TO OTHER ROBUST ESTIMATORS

In this section we compare the Minimum Sum of Distances (MSoD) estimator to other typical robust estimation schemes in the literature.

A. Iterative Trimming

Arguably, this is the simplest robust estimator. Its application involves calculating an estimate using all (noisy and corrupted) measurements, say by least squares in our case. After discarding certain number of measurements which are most inconsistent with the estimate, one recomputes the estimate using the remaining measurements. One may iterate the above process until only a predefined number of measurements remains, or until estimation error of the remaining measurements drops below some predefined level.

The main drawback of this method is that for certain corruption, the initial estimate from all the data can be made to favor some of the corrupted measurements over the uncorrupted measurements. We are not aware of any work that carefully analyzes the breakdown point of such an iterative method. However, we found that we can make this method fail using far fewer corrupted measurements than the breakdown point calculated for the MSoD estimator. Figure 3 shows a simple example in which the iterative least squares method fails but MSoD succeeds.

B. Random Sampling

Another popular approach to obtain robust estimate is through the RANdom SAMpling Consensus (RANSAC) method [9]. In our context, this corresponds to randomly selecting n of the m measurements (equations) and solving x . One then checks how many other measurements are consistent with this estimate, say error incurred is below some level. The algorithm repeatedly select sets of n measurements until an estimate with high consensus is obtained. In theory, this approach has a breakdown point of 50%.

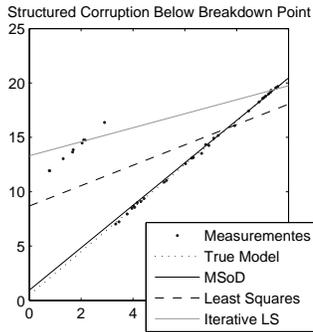


Fig. 2. We attempt to estimate a line model from which 40 noisy and corrupted points are drawn. The breakdown point of the MSoD estimator is 10 points. Corrupting the 10 leftmost points corresponds to the worst-case in which the MSoD will fail. In the example shown here we corrupted only the 9 leftmost points. Shown in the plot are the initial model estimated using least-squares for all the points, the model estimated by the iterative least-squares method, and that estimated by the MSoD. We can see that the MSoD works well, but the iterative trimming method, labeled “iterative LS,” fails to converge to a good model.

With p randomly selected sets of n measurements, the probably that at least one set contains no corrupted measurements at all is $1 - (1 - q^n)^p$ where q is the percentage of uncorrupted points. When n is small, this probability of success can be very high with relatively small number of selections – the reason why RANSAC has been very popular amongst practitioners. However, ensuring a fixed probability of success requires that the number of selections p grows *exponentially* in n , making it utterly inefficient when the dimension n is high. Linear programming solvers which minimize the MSoD cost function, on the other hand, require time polynomial in the size of the matrix A . Hence, MSoD is more scalable than RANSAC in dimension n , despite a lower breakdown point.³

VI. APPLICATION - POSITION ESTIMATION FOR A VEHICLE MOVING ON THE PLANE

In this subsection we present a “real-life” application that demonstrates the potential benefits of the Minimum Sum of Distances Estimator (MSoD). The problem which we address is estimating the position, orientation and velocity of a vehicle moving in 2D. The vehicle has inertial navigation sensors (gyroscopes) that generate noisy measurements of its velocity v and its rate of orientation change $\dot{\theta}$. In addition, the vehicle receives noisy measurements of its east, e , and north, n , position. A typical source for such measurements is a GPS system, which may produce corrupted or erroneous measurements due to multi-paths. The inertial measurements are generated every t_s seconds, while the position measurements are generated every T_s seconds, with $t_s \ll T_s$.

Given the car state at time t_0 , its position at time t_1 is

$$\begin{aligned} e(t_1) &= e(t_0) + \int_{t_0}^{t_1} \cos \theta(\tau) v(\tau) d\tau \\ n(t_1) &= n(t_0) + \int_{t_0}^{t_1} \sin \theta(\tau) v(\tau) d\tau. \end{aligned}$$

³It has been shown in the literature that for randomly generated A , the breakdown point of MSoD grows linearly in m [2], [10]. However, the fraction is normally bounded from above by 1/3.

Denote by $\hat{\cdot}$ our estimate of the car state and by $\mathbf{x} = (e - \hat{e}, n - \hat{n}, \theta - \hat{\theta}, v - \hat{v})^T$ our (presumably small) estimation error. Denote by g_e, g_n the position measurements and by $\mathbf{y} \doteq (y_0^T, \dots, y_d^T)^T$ the measurement residuals over a dT_s -time period, where

$$\mathbf{y}_k^T \doteq \begin{pmatrix} g_e(t + kT_s) - \hat{e}(t + kT_s) \\ g_n(t + kT_s) - \hat{n}(t + kT_s) \end{pmatrix}.$$

We can now write for any window size d :

$$\mathbf{y}(t) \approx A(t) \mathbf{x}(t). \quad (17)$$

The approximation is due to the linearization of the nonlinear relation between the estimation error and the measurement residuals (both assumed to be small), and due to the noise and corruptions of the position measurements.

Equation (17) is the linear model on which we apply our estimation scheme. Every time a new position measurement is generated we use it together with the last d position measurements to correct the vehicle estimated state. The matrix $A(t)$ and the estimated expected positions in the \mathbf{y} vector are regenerated every time a new position measurement arrives to reflect our best estimate so far.

Simulation results are given in Figure 4. In this simulation, the breakdown point, calculated by Algorithm 1, ranges from 4 to 6, depending on the vehicle maneuvers. While the number of corrupted measurements occasionally exceeded the breakdown point, the results were still remarkably good. This is because the breakdown point represent a worst case scenario whose probability is relatively low. For comparison we also show in Figure 4 simulation results when a standard nonlinear Kalman filter was used for this system.

VII. CONCLUSION

The main contribution of this paper was to show that the MSoD estimator, which was known to be *robust* with respect to corruption, is also *stable* with respect to noise. We also showed how to quantify the robustness and stability properties for deterministic matrices. Further study, for which the results in this paper can be used as a basis, is still needed. Key problems include developing a probabilistic or average-case analysis, as well as studying whether reweighting (by scaling the \mathbf{a}_i) can improve the estimator.

APPENDIX

A. Proof of Lemma 3.1

Proof: WLOG (Remark 3.1), we can assume $\mathbf{a}_i^T \mathbf{v} \geq 0 \forall i \in [m]$. Define $J_+ \doteq \{i \mid \mathbf{a}_i^T \mathbf{q}_{J_0} > y_i\}$, $J_- \doteq \{i \mid \mathbf{a}_i^T \mathbf{q}_{J_0} < y_i\}$. Note that J_0, J_+ and J_- are disjoint, and due to assumption 3.2, $J_0 \cup J_+ \cup J_- = [m]$. Using these definitions we can write the derivative explicitly as:

$$D_v^+ C_y(\mathbf{q}_{J_0}) = \sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{v}| + \sum_{i \in J_+} \mathbf{a}_i^T \mathbf{v} - \sum_{i \in J_-} \mathbf{a}_i^T \mathbf{v} < 0. \quad (18)$$

Consider the set:

$$\{\mathbf{w}_{i'} \mid i' \in J_0, \quad \mathbf{a}_{i'}^T \mathbf{w}_{i'} > 0, \quad \forall i \in J_0 \setminus \{i'\} : \mathbf{a}_i^T \mathbf{w}_{i'} = 0\}.$$

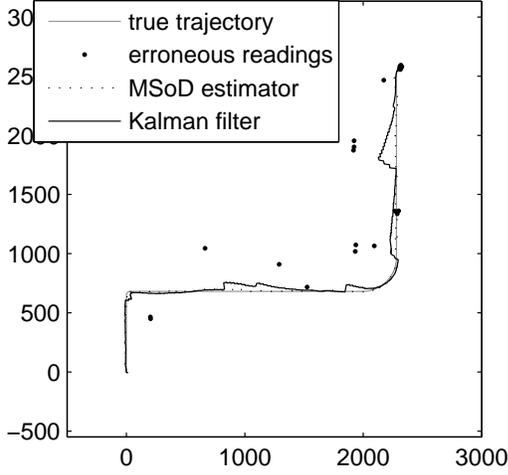


Fig. 3. Estimating a vehicle position which is moving in a 2D plane from noisy and corrupted measurements. The MSoD estimation scheme was applied on the linear model (17) using $d = 19$. The units in the plot are meters. The car average velocity is $85_{km/h}$. New position measurements are generated every $T_s = 1$ seconds. Uncorrupted position measurements have noise with 10_m standard deviation. Corrupted measurements have errors which are uniformly distributed up to 400_m . The system have a 0.06 (6%) probability of switching from an uncorrupted to a corrupted mode, and a 0.5 probability of switching from a corrupted mode to an uncorrupted mode. The maximum and the average magnitude of the position errors were 55_m and 9_m , respectively. For comparison we also show the results of using standard nonlinear Kalman filter. The standard deviation of the position errors, used to calculate the Kalman gains, was 200_m . The maximum and the average magnitude of the position errors were 157_m and 30_m , respectively.

This set contains n vectors. Each vector is parallel to all but one of the hyperplanes in J_0 . And with respect to the hyperplane to which the vector is not parallel, it points to same half space of this hyperplane as the one that \mathbf{v} points to (due to the assumption that $\mathbf{a}_i^T \mathbf{v} \geq 0$). Because there are n vectors in this set of $\mathbf{w}_{i'}$'s, they span the space, and thus we can write $\mathbf{v} = \sum_{i' \in J_0} \alpha_{i'} \mathbf{w}_{i'}$, $\alpha_{i'} \in \mathbb{R}$. Furthermore, $\forall i \in J_0$:

$$0 < \mathbf{a}_i^T \mathbf{v} = \sum_{i' \in J_0} \alpha_{i'} \mathbf{a}_i^T \mathbf{w}_{i'} = \alpha_i \mathbf{a}_i^T \mathbf{w}_i,$$

thus $\alpha_{i'} > 0 \forall i' \in J_0$. In other words, \mathbf{v} is in the cone generated by the $\mathbf{w}_{i'}$'s.

Now,

$$\sum_{i' \in J_0} \alpha_{i'} \sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{w}_{i'}| = \sum_{i' \in J_0} \alpha_{i'} \mathbf{a}_{i'}^T \mathbf{w}_{i'} = \sum_{i \in J_0} \sum_{i' \in J_0} \alpha_{i'} \mathbf{a}_i^T \mathbf{w}_{i'}$$

and thus,

$$\begin{aligned} \sum_{i' \in J_0} \alpha_{i'} \left(\sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{w}_{i'}| + \sum_{i \in J_+} \mathbf{a}_i^T \mathbf{w}_{i'} - \sum_{i \in J_-} \mathbf{a}_i^T \mathbf{w}_{i'} \right) = \\ \sum_{i \in J_0} \mathbf{a}_i^T \mathbf{v} + \sum_{i \in J_+} \mathbf{a}_i^T \mathbf{v} - \sum_{i \in J_-} \mathbf{a}_i^T \mathbf{v} < 0. \end{aligned}$$

Since the $\alpha_{i'}$'s are all positive, there must exist at least one $j \in J_0$ such that

$$\sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{w}_j| + \sum_{i \in J_+} \mathbf{a}_i^T \mathbf{w}_j - \sum_{i \in J_-} \mathbf{a}_i^T \mathbf{w}_j < 0. \quad (19)$$

Setting $\mathbf{v}' = \mathbf{w}_j$, we satisfy the first and second properties of the lemma. To show that we also satisfy the third property, note that from (19) we can write:

$$\begin{aligned} \sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{v}'| + \sum_{i \in J_+ \cap I} \mathbf{a}_i^T \mathbf{v}' - \sum_{i \in J_- \cap I} \mathbf{a}_i^T \mathbf{v}' < \\ - \sum_{i \in J_+ \cap I^c} \mathbf{a}_i^T \mathbf{v}' + \sum_{i \in J_- \cap I^c} \mathbf{a}_i^T \mathbf{v}' \leq \sum_{i \in I^c} |\mathbf{a}_i^T \mathbf{v}'| < \\ \sum_{i \in J_0} |\mathbf{a}_i^T \mathbf{v}'| + \sum_{i \in J_+ \cap I} |\mathbf{a}_i^T \mathbf{v}'| + \sum_{i \in J_- \cap I} |\mathbf{a}_i^T \mathbf{v}'|, \quad (20) \end{aligned}$$

where the last inequality follows because (11) holds. This shows that there must exist a $k \in I \setminus J_0$ such that either $k \in J_+$ and $\mathbf{a}_k^T \mathbf{v}' < 0$ or $k \in J_-$ and $\mathbf{a}_k^T \mathbf{v}' > 0$. By the definitions of J_+ and J_- this is the third property. ■

B. Proof of Lemma 3.2

Before we prove Lemma 3.2, we will prove a few preliminaries results needed in the proof of the lemma.

Proposition 1.1: Let $\tilde{\mathbf{x}} \in \mathbb{R}$ and define J_0 as the set of hyperplanes passing through $\tilde{\mathbf{x}}$. Assume that $|J_0 \cap I| < n$ and that for all $\mathbf{v} \in \mathbb{R}^n$ such that

$$\mathbf{a}_i^T \mathbf{v} = 0, \quad \forall i \in J_0 \cap I \quad (21)$$

holds,

$$\text{sgn}(\mathbf{a}_i^T \mathbf{v}) = \text{sgn}(\mathbf{a}_i^T \tilde{\mathbf{x}} - y_i), \quad \forall i \in I \setminus J_0 \quad (22)$$

does not hold. Under these assumptions, for every $\mathbf{n} \in \mathbb{R}^n$ there exists $\mathbf{q}_{J_1} = A_{J_1}^{-1} \mathbf{y}_{J_1}$, $J_1 \subset I$, $|J_1| = n$, such that

$$\mathbf{n}^T (\mathbf{q}_{J_1} - \tilde{\mathbf{x}}) \geq 0 \quad (23)$$

holds.

Proof: We give a constructive proof, showing that given any $\mathbf{v} \in \mathbb{R}^n$ such a \mathbf{q}_{J_1} can be found using Algorithm 2.

Algorithm 2 Auxiliary Algorithm for Proposition A.1

Input: $\tilde{\mathbf{x}} \in \mathbb{R}^n$, $\mathbf{n} \in \mathbb{R}^n$

1: $K_1 \leftarrow J_0 \cap I$, $K_2 = \emptyset$, $\mathbf{q} = \tilde{\mathbf{x}}$

2: **repeat**

3: $B \leftarrow A_{\{K_1 \cup K_2\}}^T$

4: $\mathbf{v}_\perp \leftarrow B (B^T B)^{-1} B^T \mathbf{n}$, $\mathbf{v}_\parallel \leftarrow \mathbf{n} - \mathbf{v}_\perp$

5: **if** $\mathbf{v}_\parallel \neq \mathbf{0}$ **then**

6: $L \leftarrow \{i \in I \mid \text{sgn}(\mathbf{a}_i^T \mathbf{v}_\parallel) = -\text{sgn}(\mathbf{a}_i^T \tilde{\mathbf{x}} - y_i)\}$

7: $i \leftarrow \arg \min_{i \in L} \frac{|\mathbf{a}_i^T \mathbf{q} - y_i|}{|\mathbf{a}_i^T \mathbf{v}_\parallel|}$

8: $\mathbf{w}_i \leftarrow \mathbf{v}_\parallel$, $\alpha_i \leftarrow \frac{|\mathbf{a}_i^T \mathbf{q} - y_i|}{|\mathbf{a}_i^T \mathbf{w}_i|}$, $\mathbf{q} \leftarrow \mathbf{q} + \alpha_i \mathbf{w}_i$

9: $K_2 \leftarrow K_2 \cup \{i\}$

10: **end if**

11: **until** $\mathbf{v}_\parallel = \mathbf{0}$

12: choose $K_3 \subset I \setminus (K_1 \cup K_2)$ s.t. $|K_3| = n - |K_1 \cup K_2|$

13: $J_1 \leftarrow K_1 \cup K_2 \cup K_3$, $\mathbf{q}_{J_1} \leftarrow A_{J_1}^{-1} \mathbf{y}_{J_1}$

Output: \mathbf{q}_{J_1} for which (23) holds

First, we show that this algorithm exits in finite time, and $|K_1 \cup K_2 \cup K_3| = n$. We will use the notation $S_K \doteq \text{span}\{\mathbf{a}_i \mid i \in K\}$. Note that lines 3 and 4 generate $\mathbf{v}_\perp \in$

$S_{K_1 \cup K_2}$ and $\mathbf{v}_{\parallel} \perp S_{K_1 \cup K_2}$. $\mathbf{v}_{\parallel} \perp S_{K_1 \cup K_2}$ implies that (21) holds when we substitute \mathbf{v} with \mathbf{v}_{\parallel} . Thus due to the assumption, the set L is nonempty. Furthermore, because $\mathbf{v}_{\parallel} \perp S_{K_1 \cup K_2}$, $i \notin K_1 \cup K_2$. Thus K_2 grows by one every iteration. Once $|K_1 \cup K_2| = n$, $S_{K_1 \cup K_2}$ equals \mathbb{R}^n and thus $\mathbf{v}_{\parallel} = 0$ and the loop exits. By definition of K_3 , which could be an empty set, $|K_1 \cup K_2 \cup K_3| = n$.

Second, we show by induction that after every time line 9 is executed \mathbf{q} is in the intersection of all the hyperplanes $K_1 \cup K_2$:

$$\mathbf{a}_i^T \mathbf{q} = y_i, \quad \forall i \in K_1 \cup K_2. \quad (24)$$

Furthermore, we will also show that no hyperplane from $I \setminus (K_1 \cup K_2)$ passes between $\tilde{\mathbf{x}}$ and \mathbf{q} :

$$\text{sgn}(\mathbf{a}_i \mathbf{q} - y_i) = \text{sgn}(\mathbf{a}_i \tilde{\mathbf{x}} - y_i), \quad \forall i \in (K_1 \cup K_2) \quad (25)$$

Before executing line 8 for the first time we have $\mathbf{q} = \tilde{\mathbf{x}}$, $K_1 = J_0$, $K_2 = \emptyset$. By definition of J_0 (24) holds and since $\mathbf{q} = \tilde{\mathbf{x}}$, (25) holds trivially. We therefore assume (24) and (25) hold before executing line 6. The set L contains all the hyperplanes which we would have crossed had we went from $\tilde{\mathbf{x}}$ in direction \mathbf{v}_{\parallel} . As mentioned above, $L \cap (K_1 \cup K_2) = \emptyset$. By the induction assumption, none of these hyperplanes passes between \mathbf{q} and $\tilde{\mathbf{x}}$, and thus this set L is exactly all the hyperplanes which we would have crossed had we went from \mathbf{q} in direction \mathbf{v}_{\parallel} . In line 7 we choose the closest hyperplane in direction \mathbf{v}_{\parallel} . Moving from \mathbf{q} to $\mathbf{q} + \alpha_i \mathbf{w}_i$ will bring us to this hyperplane without crossing any other hyperplane. Since $\mathbf{w}_i = \mathbf{v}_{\parallel} \notin S_{K_1 \cup K_2}$, and using the induction assumption that (24) holds, this new point must be on the intersection of hyperplanes $K_1 \cup K_2$.

Third, and last, we show that (23) holds. When the algorithm ends, we have that both \mathbf{q}_{J_1} and \mathbf{q} are on hyperplanes $K_1 \cup K_2$. Thus, $\mathbf{a}_i^T \mathbf{r} = 0 \forall i \in K_1 \cup K_2$ where $\mathbf{r} \doteq \mathbf{q}_{J_1} - \mathbf{q}$. Since the algorithm ends when $\mathbf{v}_{\parallel} = 0$ we also have $\mathbf{n} \in S_{K_1 \cup K_2}$. As a result, $\mathbf{n}^T (\mathbf{q}_{J_1} - \tilde{\mathbf{x}}) = \mathbf{n}^T (\sum_{i \in K_2} \mathbf{w}_i + \mathbf{r}) = \sum_{i \in K_2} \mathbf{n}^T \mathbf{w}_i$. Since each \mathbf{w}_i is a component in an orthogonal decomposition of \mathbf{n} , we get that (23) must hold. ■

Corollary 1.1: For $\tilde{\mathbf{x}} \in \mathbb{R}$ and J_0 defined as the set of hyperplanes passing through $\tilde{\mathbf{x}}$, if $\tilde{\mathbf{x}} \notin P_I$ then $|J_0 \cap I| < n$ and there exists $\mathbf{v} \in \mathbb{R}$ such that both (21) and (22) hold.

Proof: If $\tilde{\mathbf{x}} \notin P_I$ then necessarily $|J_0 \cap I| < n$. The polytope P_I is the convex hull of the points in Q_I (10). The point $\tilde{\mathbf{x}}$ will be outside of P_I if and only we can find a closed half space whose edge contains $\tilde{\mathbf{x}}$ but no point from Q_I is contained in the closed half space. If there does not exist $\mathbf{v} \in \mathbb{R}$ for which both (21) and (22) hold then by Proposition A.1, every closed half space containing $\tilde{\mathbf{x}}$ in its edge ($\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{n}^T (\mathbf{x} - \tilde{\mathbf{x}}) \geq 0\}$) also contains a point from Q_I by (23), implying $\tilde{\mathbf{x}} \in P_I$. ■

Proof of Lemma 3.2: Let $\tilde{\mathbf{x}}$ be the point outside P_I that minimizes the cost function. Since it is the minimum we know that for each $\mathbf{v} \in \mathbb{R}^n$, $D_{\mathbf{v}}^+ C_{\mathbf{y}}(\tilde{\mathbf{x}}) \geq 0$. Note that the derivative depends on \mathbf{y} only through $\text{sgn}(\mathbf{y} - A\tilde{\mathbf{x}})$. Thus taking any hyperplane which does not pass through $\tilde{\mathbf{x}}$ away from $\tilde{\mathbf{x}}$, will not affect the derivatives from $\tilde{\mathbf{x}}$. This way we

can move away from P_I any corrupted hyperplane which does not pass through $\tilde{\mathbf{x}}$. Next we show that we can also move the corrupted hyperplanes that do pass through $\tilde{\mathbf{x}}$ (as well as $\tilde{\mathbf{x}}$ as the point minimizing the cost function) away from P_I .

In the statement of Lemma 3.2 we assumed $\tilde{\mathbf{x}} \notin P_I$. Thus, by Corollary A.1 there exists $\mathbf{v} \in \mathbb{R}^n$ such that (21) and (22) hold. Note that if (22) holds for some \mathbf{v} then it also holds for all \mathbf{v} in some open neighborhood of \mathbf{v} . Since we assume the hyperplanes are in general direction (Assumption 3.1), we can therefore choose \mathbf{v} so that $\mathbf{a}_i^T \mathbf{v} \neq 0 \forall i \in J_0 \setminus I$ while keeping (21) true. Now, assume the cost function is minimized at the intersection of some hyperplanes. If we move these hyperplanes without having the intersection cross any other hyperplane, the cost function will still be minimized at the (modified) intersection of these hyperplane. Condition (21) implies that the direction \mathbf{v} from $\tilde{\mathbf{x}}$ is contained in the uncorrupted hyperplanes that passes through $\tilde{\mathbf{x}}$ while Condition (22) implies that this direction is pointing away from any of the remaining uncorrupted hyperplanes. Thus the existence of a direction \mathbf{v} for which (21) and (22) hold and also $\mathbf{a}_i^T \mathbf{v} \neq 0 \forall i \in J_0 \setminus I$ implies we can synchronously move all the hyperplanes $J_0 \setminus I$, and no matter how far we move, the intersection J_0 will not cross any of the hyperplanes $I \setminus J_0$. Moving far enough we can eventually take all the hyperplanes $J_0 \setminus I$ outside P_I . ■

REFERENCES

- [1] H. P. Lopuha and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *Annals of Statistics*, vol. 19, no. 1, pp. 229–248, 1991.
- [2] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [3] E. Candes and P. A. Randall, "Highly robust error correction by convex programming," *preprint*, <http://arxiv.org/abs/cs.IT/0612124>, 2006.
- [4] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm near-solution approximates the sparsest near-solution," *Communication on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 907–934, March 2006.
- [5] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, pp. 267–288, 1996.
- [6] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *to appear in Annals of Statistics*, 2006.
- [7] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *preprint*, <http://arxiv.org/abs/0803.2392v2>, 2008.
- [8] D. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," *IEEE. Trans. Info. Theory*, 2006.
- [9] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.
- [10] D. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," *preprint*, <http://www.math.utah.edu/tanner/>, 2007.