Sparse Total Least Squares: Analysis and Greedy Algorithms

Bogdan Dumitrescu

Tampere International Center for Signal Processing, Tampere University of Technology, Finland, and Department of Automatic Control and Computers, "Politehnica" University of Bucharest, Romania. E-mail: bogdand@cs.tut.fi.

Abstract

Linear systems where both the matrix and the right hand side are affected by noise or uncertainties are typically solved via a total least squares (TLS) approach. We study the *sparse* TLS problem, with solutions having s nonzero elements. We present a set of conditions under which the sparse least squares (LS) and TLS problems have the same support. The conditions are expressed in terms of the restricted isometry constants and the minimum nonzero principal angle between subspaces made of s columns of the matrix. We also study several greedy algorithms for solving the sparse TLS problem. Inspired by our theoretical results, we infer that an effective approach is to compute the support of the LS solution, then solve the standard TLS problem restrained to that support. Indeed, doing so with orthogonal matching pursuit (OMP) gives consistently good results. Another greedy strategy is to extend the support with the column that minimizes the smallest angle with the subspace (SAS) corresponding to the already chosen support, which is a combined LS-TLS heuristic. We show via simulations that SAS gives the best results for small problem sizes and high and medium signal-to-noise ratio (SNR), while OMP is best or nearly best in almost all cases.

Keywords: Total least squares, sparse solutions, greedy algorithms, orthogonal matching pursuit

AMS classification: 15A06

1. Introduction

Consider the linear system $\boldsymbol{Ax} = \boldsymbol{b}$, with $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, known to have a sparse solution, that is one with few nonzero elements. This problem has been investigated in hundreds of papers in the last decade, especially in signal processing literature, where $\boldsymbol{b}$ is often called observation or measurements vector and $\boldsymbol{A}$ is the dictionary or the measurement matrix. In a noisy environment, when the equality cannot be satisfied exactly by any sparse vector $\boldsymbol{x}$, the model is usually changed to $\boldsymbol{Ax} = \boldsymbol{b} + \boldsymbol{f}$, where $\boldsymbol{f}$ is the measurement noise; hence, the typical approach is to find a least-squares (LS) sparse solution. However, there are applications where the measurement matrix is also affected by noise or uncertainties and where a more natural model is the total least-squares (TLS) one. There are few papers dedicated to sparse TLS or similar problems. The first specialized algorithm seems to appear in [1]. A related paper is [2], where an error analysis of the Basis Pursuit algorithm (which computes an LS solution) is presented for the TLS setup, with both $\boldsymbol{A}$ and $\boldsymbol{b}$ affected by noise. A similar analysis for greedy algorithms, in particular for Compressive Sampling Matching Pursuit (CoSaMP), was presented in [3]; Orthogonal Matching Pursuit (OMP) is analyzed in this context in [4], with emphasis on the conditions under which the support of the solution is recovered. The dictionary can be inadequate due not only to noise, but to discretization issues; this aspect is termed basis mismatch and analyzed in [5] and [6]. A very recent work is [7], with algorithms that are especially efficient for structured perturbations.

The TLS solution of the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is given by

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},\boldsymbol{E},\boldsymbol{f}} & \|[\boldsymbol{E}\ \boldsymbol{f}]\|_F \\ \text{s.t.} & (\boldsymbol{A}+\boldsymbol{E})\boldsymbol{x} = \boldsymbol{b}+\boldsymbol{f} \end{array} \tag{1}$$

The solution results by taking $[\boldsymbol{E}\ \boldsymbol{f}]$ as the matrix with smallest Frobenius norm that added to $[\boldsymbol{A}\ \boldsymbol{b}]$ makes the result singular. As such, it is related to the smallest singular value of $[\boldsymbol{A}\ \boldsymbol{b}]$: at optimality we have $\|[\boldsymbol{E}\ \boldsymbol{f}]\|_F = \sigma_{\min}([\boldsymbol{A}\ \boldsymbol{b}])$.

We consider the sparse version of (1), searching sparse TLS solutions $\boldsymbol{x} \in \mathbb{R}^n$ having s nonzero elements ($\|\boldsymbol{x}\|_0 = s$). In [1], sparsity is promoted by replacing the criterion of (1) with $\|[\boldsymbol{E}\ \boldsymbol{f}]\|_F^2 + \lambda\|\boldsymbol{x}\|_1$, where $\lambda > 0$ is a constant. For LS problems, this is a convex relaxation technique, but in the TLS case the new problem is not convex and an algorithm based on

alternating coordinate descent is proposed in [1]. We focus here on greedy algorithms, due to their absence from the literature and for other reasons that will be later detailed.

In principle, the sparse TLS problem can be solved in two steps. This does not make the problem simpler, but it can be a platform for greedy approaches and also other algorithms.

1. *Selection of nonzero elements.* If $\mathcal{I}$ is a set of s indices (the support), then denote $\boldsymbol{x}_{\mathcal{I}} \in \mathbb{R}^s$ the vector of nonzero elements of $\boldsymbol{x}$ (and assume that the other elements are zero) and $\boldsymbol{A}_{\mathcal{I}} \in \mathbb{R}^{m \times s}$ the matrix formed by the columns of $\boldsymbol{A}$ with indices in $\mathcal{I}$. The support of the sparse TLS solution with s nonzeros is given by

$$\begin{array}{ll} \min\limits_{\mathcal{I}} & \sigma_{\min}([\boldsymbol{A}_{\mathcal{I}} \ \boldsymbol{b}]) \\ \text{s.t.} & |\mathcal{I}| = s \end{array} \tag{2}$$

This problem is hard and its solution can be generally found only by enumeration; this is where greedy algorithms apply as a practical approach.

2. *TLS solution calculation.* Once the support is determined, the solution $\boldsymbol{x}_{\mathcal{I}}$ results from solving the standard TLS problem (1) with $\boldsymbol{A}_{\mathcal{I}}$ instead of $\boldsymbol{A}$. The well known algorithm based on SVD can be found e.g. in [8], [9].

Our contributions are twofold. First, in Section 2, we present a set of conditions under which the sparse LS and TLS solutions have the same support. This result complements those from other papers like [2, 3, 4], stating conditions that depend only on the data and not on the algorithm for solving the LS or TLS problem. The theoretical conditions we present suggest that it may not matter much, especially when the noise level is small, if we find the support of the sparse TLS solution using a sparse LS method (from the wealth of available ones) or a dedicated TLS method. (Interestingly, a similar conclusion is reached via probabilistic arguments in [7].)

Second, we investigate methods that follow the two-step scheme advocated above for solving the sparse TLS problem, focusing on greedy algorithms. Due to its simplicity and speed, a first option is Orthogonal Matching Pursuit (OMP), shown to have a robust behavior. Section 3 presents other greedy TLS algorithms; one is based on the method from [10] for computing the restricted isometry property (RIP) constants; another algorithm is our contribution and, in the greedy process of column selection for building the support $\mathcal{I}$, chooses the column that makes the smallest angle with the subspace made of the already selected columns and the vector $\boldsymbol{b}$ (or, equivalently, has the largest projection on this subspace). Section 4 is dedicated

to simulations that allow comparisons between our greedy algorithms and that from [1] and show that greedy algorithms are better in many situations, especially for high sparsity and not very low signal to noise ratio (SNR).

2. When do sparse LS and TLS solutions have the same support ?

Let us consider the sparse LS problem, defined again for fixed sparsity s by

$$\begin{array}{ll} \min\limits_{\mathcal{I},\boldsymbol{x}} & \|\boldsymbol{b}-\boldsymbol{A}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}}\| \\ \text{s.t.} & |\mathcal{I}|=s \end{array} \tag{3}$$

When $\boldsymbol{b} \in \mathrm{Im}\boldsymbol{A}_{\mathcal{I}}$ for some $\mathcal{I}$ with $|\mathcal{I}| = s$, the LS and TLS solutions coincide. It is reasonable to expect that, if $\boldsymbol{b}$ is near enough to a subspace $\mathrm{Im}\boldsymbol{A}_{\mathcal{I}}$, the supports of the LS and TLS solutions are the same. We aim here to give a precise bound for this to happen.

2.1. Assumptions and definitions

We restrain the class of LS solutions by examining only vectors that satisfy the condition

$$|x_i| \geq \beta\|\boldsymbol{x}\|, \quad i \in \mathcal{I}, \tag{4}$$

where β is a constant. Otherwise, the distance between vectors with different supports can be arbitrarily small. A similar condition, but in absolute form, appears in the analysis of sparse recovery with OMP in [11]; even more restrictive conditions are present in [4]. We assume that the matrix $\boldsymbol{A}$ satisfies RIP [12] with constants $\delta_s^{\min}$ and $\delta_s^{\max}$, i.e.

$$(1-\delta_s^{\min})\|\boldsymbol{x}\|^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|^2 \leq (1+\delta_s^{\max})\|\boldsymbol{x}\|^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \ \|\boldsymbol{x}\|_0 = s. \tag{5}$$

For any set $\mathcal{I}$ with $|\mathcal{I}| = s$, this implies

$$\sigma_{\min}^2(\boldsymbol{A}_{\mathcal{I}}) \geq 1-\delta_s^{\min}, \quad \sigma_{\max}^2(\boldsymbol{A}_{\mathcal{I}}) \leq 1+\delta_s^{\max}. \tag{6}$$

We assume that $\delta_{2s}^{\min} < 1$, which means that any $2s$ columns of $\boldsymbol{A}$ are linearly independent. This is not a restrictive condition. In the generic case where all the submatrices of $\boldsymbol{A}$ with m columns are nonsingular, it is enough to have $s \leq m/2$.

We will assume implicitly (see Lemma 3) that the solution of the sparse LS problem (3) is unique, i.e. there are no optimal solutions with different supports; this is the generic case.

Similarly, we assume that the solution of the TLS sparse problem (2) exists and is unique. This is again a generic situation. Unlike the LS problem, even if the support $\mathcal{I}$ is given, the TLS problem $\boldsymbol{A}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}} = \boldsymbol{b}$ may have multiple or no solution. This can happen only if [9, p.38] the columns of $\boldsymbol{A}_{\mathcal{I}}$ are linearly dependent, case already dismissed above, or $\boldsymbol{b}$ is orthogonal on the columns of $\boldsymbol{A}_{\mathcal{I}}$, and hence on the columns of $\boldsymbol{A}$, which is not only unlikely, but makes the linear model $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ completely inappropriate.

Let θ_s be the minimum smallest nonzero principal angle made by the subspaces generated by any two distinct (but not necessarily disjoint) subsets of s columns of $\boldsymbol{A}$, as further explained. Let $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{R}^{m\times s}$ be two submatrices of $\boldsymbol{A}$ with s columns, $\mathcal{S}_1 = \text{Im}\boldsymbol{A}_1$, $\mathcal{S}_2 = \text{Im}\boldsymbol{A}_2$ the subspaces they generate and $\mathcal{S}$ their intersection. Let $\mathcal{S}_{o1}$, $\mathcal{S}_{o2}$ be subspaces orthogonal on $\mathcal{S}$, such that $\mathcal{S}_1 = \mathcal{S} \cup \mathcal{S}_{o1}$, $\mathcal{S}_2 = \mathcal{S} \cup \mathcal{S}_{o2}$. The smallest nonzero principal angle between $\mathcal{S}_1$ and $\mathcal{S}_2$ is

$$\alpha_{12} = \arccos \max_{\boldsymbol{v}_1\in\mathcal{S}_{o1}, \boldsymbol{v}_2\in\mathcal{S}_{o2}} \frac{|\boldsymbol{v}_1^T\boldsymbol{v}_2|}{\|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\|}. \tag{7}$$

The minimum α_{12} over all possible pairs $\mathcal{S}_1$, $\mathcal{S}_2$ is θ_s. Since it will often appear in our results, we denote

$$\tau_s = \tan\frac{\theta_s}{2}. \tag{8}$$

Using the above subspaces, we decompose vector $\boldsymbol{b}$ as in Figure 1. Let $\boldsymbol{b} = \boldsymbol{b}_c + \boldsymbol{b}_o$, with $\boldsymbol{b}_c \in \mathcal{S}$ and $\boldsymbol{b}_o \perp \mathcal{S}$. Let $\boldsymbol{b}_{o1}$ be the projection of $\boldsymbol{b}_o$ on $\mathcal{S}_1$ (in the figure, $\mathcal{S}_1$ is the extension of the front face of the rectangular parallelepiped) and denote α_1 the angle between these two vectors; note that $\boldsymbol{b}_{o1} \in \mathcal{S}_{o1}$. Denote $\boldsymbol{b}_1$ the projection of $\boldsymbol{b}$ on $\mathcal{S}_1$; it follows that $\boldsymbol{b}_1 = \boldsymbol{b}_c + \boldsymbol{b}_{o1}$. The residual of the overdetermined linear system $\boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{b}$ can be hence expressed as

$$\boldsymbol{r}_1 = \boldsymbol{b} - \boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{b} - \boldsymbol{b}_1 = \boldsymbol{b}_o - \boldsymbol{b}_{o1}. \tag{9}$$

Similar quantities are defined for $\mathcal{S}_2$, using the same notations, with subscript 2 instead of 1.

2.2. Technical results

Using the above definitions, we give some preparatory results. Remind that we consider only subspaces generating solutions for which condition (4) holds. First, we express the bound on the elements of the LS solution as a function of $\boldsymbol{b}$.

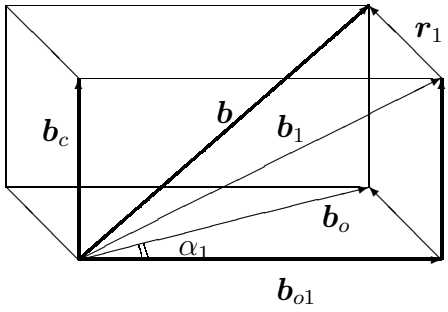

Figure 1: Decomposition of vector $\boldsymbol{b}$.

Lemma 1. *The elements of the LS solution to the system $\boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{b}$ satisfy the bound*

$$|x_i| \geq \frac{\beta\|\boldsymbol{b}_1\|}{\sqrt{1+\delta_s^{\max}}}, \quad \forall i = 1:s. \tag{10}$$

Proof: Since $\boldsymbol{b}_1$ is the projection of $\boldsymbol{b}$ on $\mathcal{S}_1$, it follows that the optimal $\boldsymbol{x}_1$ satisfies $\boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{b}_1$. It results that $\|\boldsymbol{A}_1\| \cdot \|\boldsymbol{x}_1\| \geq \|\boldsymbol{b}_1\|$ and, using (6), $\|\boldsymbol{x}_1\| \geq \|\boldsymbol{b}_1\|/\sqrt{1+\delta_s^{\max}}$. Combining this with (4) gives (10). ■

Lemma 2. *For any $\mathcal{S}_2$, the component of $\boldsymbol{b}$ on $\mathcal{S}_{o1}$ satisfies the bound*

$$\|\boldsymbol{b}_{o1}\| \geq \beta\|\boldsymbol{b}_1\|\sqrt{\frac{1-\delta_s^{\min}}{1+\delta_s^{\max}}}. \tag{11}$$

Proof: Let $\boldsymbol{A}_c \in \mathbb{R}^{m\times\nu}$ be the matrix made of the columns that belong to both $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. Since $\delta_{2s}^{\min} < 1$, and hence $\delta_i^{\min} < 1$, $\forall i = 1:2s$, the common subspace $\mathcal{S}$ can be generated only by the columns of $\boldsymbol{A}_c$. We write $\boldsymbol{A}_1 = [\boldsymbol{A}_c \ \tilde{\boldsymbol{A}}_1]$, with $\tilde{\boldsymbol{A}}_1 \in \mathbb{R}^{m\times(s-\nu)}$ containing the columns that belong to $\boldsymbol{A}_1$ but not to $\boldsymbol{A}_2$ (no generality is lost if the columns are permuted). This matrix can be expressed as $\tilde{\boldsymbol{A}}_1 = \boldsymbol{A}_{c1} + \boldsymbol{A}_{o1}$, with the columns of $\boldsymbol{A}_{c1}$ belonging to $\mathcal{S}$ and those of $\boldsymbol{A}_{o1}$ to $\mathcal{S}_{o1}$. We can write $\boldsymbol{A}_{c1} = \boldsymbol{A}_c\boldsymbol{Z}$ and obtain

$$\boldsymbol{A}_{o1} = \tilde{\boldsymbol{A}}_1 - \boldsymbol{A}_{c1} = [\boldsymbol{A}_c \ \tilde{\boldsymbol{A}}_1]\begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{I} \end{bmatrix} = \boldsymbol{A}_1\begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{I} \end{bmatrix}.$$

We can also write

$$\boldsymbol{b}_1 = \boldsymbol{A}_1\boldsymbol{x}_1 = [\boldsymbol{A}_c \ \tilde{\boldsymbol{A}}_1]\begin{bmatrix} \boldsymbol{x}_{c1} \\ \boldsymbol{x}_{o1} \end{bmatrix} = \underbrace{\boldsymbol{A}_c(\boldsymbol{x}_{c1} + \boldsymbol{Z}\boldsymbol{x}_{o1})}_{\boldsymbol{b}_c} + \underbrace{\boldsymbol{A}_{o1}\boldsymbol{x}_{o1}}_{\boldsymbol{b}_{o1}}.$$

Using the above, we finally get

$$\|\boldsymbol{b}_{o1}\| = \|\boldsymbol{A}_{o1}\boldsymbol{x}_{o1}\| = \left\|\boldsymbol{A}_1 \begin{bmatrix} -\boldsymbol{Z}\boldsymbol{x}_{o1} \\ \boldsymbol{x}_{o1} \end{bmatrix}\right\| \geq \sigma_{\min}(\boldsymbol{A}_1)\|\boldsymbol{x}_{o1}\| \geq \sqrt{1-\delta_s^{\min}}\|\boldsymbol{x}_{o1}\|.$$

The lowest bound is obtained when $\boldsymbol{x}_{o1}$ has a single element (when $\nu = s-1$); combining it with (10) gives (11). ■

The minimum angle between subspaces can be used to (a posteriori) guarantee the optimality of a sparse LS solution.

Lemma 3. *Denote $\boldsymbol{r}_1$ the residual of the LS problem defined by $\boldsymbol{A}_1$ and $\boldsymbol{b}$, as in (9). If*

$$\frac{\|\boldsymbol{r}_1\|}{\|\boldsymbol{b}_{o1}\|} < \tau_s \tag{12}$$

for any $\mathcal{S}_2$, then the support $\mathcal{I}$ defining $\boldsymbol{A}_1 = \boldsymbol{A}_{\mathcal{I}}$ corresponds to the s-sparse LS solution of (3).

Proof: The result is geometrically intuitive. From our definitions (see again Figure 1), it results that $\tan\alpha_1 = \|\boldsymbol{r}_1\|/\|\boldsymbol{b}_{o1}\|$. If (12) holds, then $\alpha_1 < \theta_s/2$.

It is also visible that $\alpha_1 + \alpha_2 \geq \alpha_{12}$, where α_{12} is the angle (7) between $\mathcal{S}_1$ and $\mathcal{S}_2$, for any other subspace $\mathcal{S}_2$; equality is possible only if $\boldsymbol{b}_o$, $\boldsymbol{b}_{o1}$ and $\boldsymbol{b}_{o2}$ are in the same plane and the angle between $\mathcal{S}_1$ and $\mathcal{S}_2$ is the same as the angle between $\boldsymbol{b}_{o1}$ and $\boldsymbol{b}_{o2}$. Since $\alpha_{12} \geq \theta_s$, it results that $\alpha_2 > \theta_s/2 > \alpha_1$, which implies

$$\frac{\|\boldsymbol{r}_1\|}{\|\boldsymbol{b}_{o1}\|} < \frac{\|\boldsymbol{r}_2\|}{\|\boldsymbol{b}_{o2}\|}. \tag{13}$$

Since $\boldsymbol{b} = \boldsymbol{b}_c + \boldsymbol{b}_{o1} + \boldsymbol{r}_1$ and these three components are orthogonal on each other, it follows that $\|\boldsymbol{b}\|^2 = \|\boldsymbol{b}_c\|^2 + \|\boldsymbol{b}_{o1}\|^2 + \|\boldsymbol{r}_1\|^2$. Decomposing similarly on $\mathcal{S}_2$, we obtain the equality

$$\|\boldsymbol{b}_{o1}\|^2 + \|\boldsymbol{r}_1\|^2 = \|\boldsymbol{b}_{o2}\|^2 + \|\boldsymbol{r}_2\|^2. \tag{14}$$

Combining it with (13) we get $\|\boldsymbol{r}_1\| < \|\boldsymbol{r}_2\|$, which concludes the proof. ■

2.3. Main result

We are now in position to state the main result. Let us define

$$\sqrt{\gamma_s} = \beta\tau_s \left[\frac{1-\delta_s^{\min}}{(1+\tau_s^2)(1+\delta_s^{\max})\left(1+\frac{\|\boldsymbol{b}\|^2}{1-\delta_s^{\min}}\right)}\right]^{\frac{1}{2}}. \tag{15}$$

Theorem 4. *If the residual (9) of the s-sparse LS problem (3) defined by* $\boldsymbol{A}_{\mathcal{I}} = \boldsymbol{A}_1$ *and* $\boldsymbol{b}$ *satisfies the condition*

$$\frac{\|\boldsymbol{r}_1\|}{\|\boldsymbol{b}\|} \leq \sqrt{\frac{\gamma_s}{1+\gamma_s}}, \tag{16}$$

then the solutions of the s-sparse LS and TLS problems (3) and (2), respectively, have the same support.

Proof: We first note that (16) is equivalent to

$$\frac{\|\boldsymbol{r}_1\|^2}{\|\boldsymbol{b}_1\|^2 + \|\boldsymbol{r}_1\|^2} \leq \frac{\gamma_s}{1+\gamma_s} \quad \Leftrightarrow \quad \frac{\|\boldsymbol{r}_1\|^2}{\|\boldsymbol{b}_1\|^2} \leq \gamma_s.$$

Using (11), this implies

$$\frac{\|\boldsymbol{r}_1\|}{\|\boldsymbol{b}_{o1}\|} \leq \frac{\sqrt{\gamma_s}}{\beta\sqrt{\frac{1-\delta_s^{\min}}{1+\delta_s^{\max}}}} = \frac{\tau_s}{(1+\tau_s^2)^{\frac{1}{2}}\left(1+\frac{\|\boldsymbol{b}\|^2}{1-\delta_s^{\min}}\right)^{\frac{1}{2}}}. \tag{17}$$

Since the denominator of the r.h.s. of (17) is greater than 1, the condition (12) is satisfied and so Lemma 3 says that the residual $\boldsymbol{r}_1$ corresponds to the s-sparse solution of the LS problem (3).

For assessing the TLS optimality, we employ the following relation [9, Th.6.10], holding for arbitrary $\boldsymbol{A}_1$ and $\boldsymbol{b}$ (see also [13] for some improved, but more complicated bounds):

$$\sigma_{\min}^2([\boldsymbol{A}_1\ \boldsymbol{b}]) \leq \|\boldsymbol{r}_1\|^2 \leq \sigma_{\min}^2([\boldsymbol{A}_1\ \boldsymbol{b}]) \cdot \frac{\sigma_{\min}^2(\boldsymbol{A}_1) + \|\boldsymbol{b}\|^2 - \sigma_{\min}^2([\boldsymbol{A}_1\ \boldsymbol{b}])}{\sigma_{\min}^2(\boldsymbol{A}_1)}. \tag{18}$$

Combining (17) with the first inequality from (18) we obtain

$$\sigma_{\min}([\boldsymbol{A}_1\ \boldsymbol{b}]) \leq \frac{\tau_s\|\boldsymbol{b}_{o1}\|}{(1+\tau_s^2)^{\frac{1}{2}}\left(1+\frac{\|\boldsymbol{b}\|^2}{1-\delta_s^{\min}}\right)^{\frac{1}{2}}}. \tag{19}$$

Take now another support of size s, corresponding to a matrix $\boldsymbol{A}_2$. Using the second inequalities from (6) and (18) for $\boldsymbol{A}_2$, it results that

$$\sigma_{\min}([\boldsymbol{A}_2\ \boldsymbol{b}]) \geq \frac{\|\boldsymbol{r}_2\|}{\left(1+\frac{\|\boldsymbol{b}\|^2}{1-\delta_s^{\min}}\right)^{\frac{1}{2}}}. \tag{20}$$

Denote q^2 the quantity from (14). Since $\boldsymbol{A}_2$ is not LS optimal, it results from Lemma 3 that

$$\|\boldsymbol{r}_2\|^2 > \tau_s^2 \|\boldsymbol{b}_{o2}\|^2 = \tau_s^2 (q^2 - \|\boldsymbol{r}_2\|^2) \tag{21}$$

and so, since (14) also implies $q \geq \|\boldsymbol{b}_{o1}\|$,

$$\|\boldsymbol{r}_2\| > \frac{\tau_s q}{(1+\tau_s^2)^{\frac{1}{2}}} \geq \frac{\tau_s \|\boldsymbol{b}_{o1}\|}{(1+\tau_s^2)^{\frac{1}{2}}}. \tag{22}$$

Combining this inequality and (20), and comparing with (19), it results that

$$\sigma_{\min}([\boldsymbol{A}_2 \ \boldsymbol{b}]) > \sigma_{\min}([\boldsymbol{A}_1 \ \boldsymbol{b}])$$

and hence $\mathcal{I}$ is the support of the solution of the TLS problem (2). ■

2.4. Comments and some numerical evidence

The value of the bound (16) depends on the good separation between the columns of $\boldsymbol{A}$, expressed via the minimum angle θ_s and the RIP constants, and the bound β for the components of the solution. The bound (16) could be used as an optimality certificate for a computed sparse solution, once its support has been determined. Of course, estimating θ_s and the RIP constants is a difficult problem, harder than solving the LS or TLS problems. The RIP constants can be estimated with the algorithms from [10] and [14]; see also [15] for theoretical estimates for random matrices. Since the angle θ_s seems to be used here for the first time in this context, we are not aware of an algorithm for its estimation. So, the bound has mostly qualitative significance.

To estimate its tightness, we have evaluated the numerical value of the ratio between the two sides of (16),

$$\rho = \frac{\|\boldsymbol{r}_1\|}{\|\boldsymbol{b}\|} \sqrt{\frac{1+\gamma_s}{\gamma_s}}, \tag{23}$$

for systems for which the s-sparse LS and TLS solutions have *different* support (hence $\rho > 1$). If ρ is close to 1, we can conclude that the bound is tight. Due to the complexity of the computations, we considered only the case $s = 2$. We generated random matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with normally distributed entries with variance $1/m$, for each computing τ_s and the RIP constants by exhaustive search. For each $\boldsymbol{A}$, we randomly generated many vectors $\boldsymbol{b}$; for each $\boldsymbol{b}$, we took β from (4) as the minimum ratio $|x_i|/\|\boldsymbol{x}\|$ among all LS solutions $\boldsymbol{x}$ with support of size 2. (On one side, this may

be an overestimation, since some of these solutions give a large residual and hence are not meaningful candidates for the optimum; on the other side, it is hard to recommend a priori a sensible value for β.) When promising values of ρ were found, we successively "optimized" $\boldsymbol{b}$ by random search in its neighborhood.

We start by examining matrices with the minimum number of columns that can give meaningful results for $s = 2$, namely $n = 3$. With the above rudimentary optimization, the smallest value for the ratio (23) was $\rho = 1.38$, with $m = 8$; many other values below 2 were obtained for several $m \leq 8$. We go then to $n = 4$, the smallest number of columns for which the sparse LS and TLS solution could have disjoint supports. For $m = 4$ (remind that the condition $\delta_{2s}^{\min} < 1$ requires $m \geq 2s$), we have easily obtained values of ρ around 3, the minimum being 2.42. If m is increased, even lower values can result; for $m = 8$, $n = 4$, the minimum was 2.02. When $m < n$, the ratio increases; for example, for 4×5 matrices, the best value was $\rho = 5.52$. We also experimented with matrices of size $m \times 2m$, taking $A = [I\ \Phi]$, with Φ random and orthogonal; for 4×8 matrices, the best ratio was $\rho = 28.4$ while for 8×16 matrices we obtained $\rho = 54.1$. While these values suggest that the bound may be reasonably tight, it is difficult to evaluate if the values of ρ scale nicely as m, n and s grow. (The increase shown by our experiments could be due to the difficulty of the search as the dimensions grow.)

Also, as it typically happens in sparse problems, the bound (16) is probably conservative in average, especially when the denominator of (15) is large; however, it suggests that sparse LS and TLS solutions might often have the same support. To have some numerical evidence, we randomly generated $\boldsymbol{A}$ as described above. We took "true" solutions $\boldsymbol{x} \in \mathbb{R}^n$ with s nonzero normally distributed elements, obeying (4) with $\beta = 0.01$; $\boldsymbol{x}$ will be named generating solution, since it is used to build the data. The corresponding right hand side of the system is $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$. The TLS data are generated as $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{E}$ and $\hat{\boldsymbol{b}} = \boldsymbol{b} + \boldsymbol{f}$, with $\boldsymbol{E}$ and $\boldsymbol{f}$ having normally distributed elements with variance ς^2/m. So, the SNR for each data element is $-20 \log_{10} \varsigma$. For each $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{b}}$, we have solved the s-sparse LS and TLS problems by exhaustive search; hence, the optimality of the solution is guaranteed. For each considered m, n, s, we have solved 100 different problems. Due to the complexity of the computations, the data are limited to those presented in Table 1, where we give the number of problems for which the LS and TLS solutions have the same support. Naturally, the number of cases where the support of the LS and TLS solutions is the same decreases at higher noise.

Table 1: Number of cases (out of 100) where the LS and TLS solutions have the same support, for $m = 20$.

	$n = 40$				$n = 60$			
SNR $\setminus$ s	2	3	4	5	2	3	4	5
30 dB	99	100	100	98	100	100	100	98
20 dB	98	100	97	94	100	96	91	92
10 dB	86	73	50	39	85	65	43	27

However, for an SNR of 20 dB, which is realistic for many applications, the LS and TLS supports coincide in more than 90% of the problems. We repeated the experiments for $\beta = 0.05$ and the results are not significantly different from those in Table 1.

A noteworthy remark is that the optimal LS and TLS solutions have relatively often a different support than the generating solution. For an SNR of 20 or 10 dB, the LS and TLS solutions have the same support more often than one of them and the generating solution have the same support. This means that the noise is large enough to change the support of the solution.

3. Greedy algorithms for the TLS problem

The previous section is an invitation to explore the behavior of a simple sparse TLS algorithm following the two-step approach advocated in the introduction: solve the sparse LS problem (3) instead of the TLS counterpart (2) and use its support $\mathcal{I}$ to compute the sparse TLS solution. Among the many algorithms available for solving (3) we have chosen Orthogonal Matching Pursuit (OMP) and Orthogonal Least Squares (OLS) [16] (named also LS-OMP, see [17, ch.3]), due to their simplicity and greedy nature. Another recent argument is the analysis from [4], which presents conditions for OMP to recover the support of the LS solution when both $\boldsymbol{A}$ and $\boldsymbol{b}$ are perturbed; they are quite different from those of Theorem 4 and their conservativeness is also not estimated, but combined they would show when OMP recovers the support of the TLS solution. It is possible that other sparse LS algorithms are superior, but OMP and OLS are sufficient to our purposes.

Although this approach uses off-the-shelf tools and gives good results (as shown in the next section), we consider necessary to investigate also

Table 2: Acronyms for the sparse TLS algorithms.

OMP	Orthogonal Matching Pursuit (standard)
OLS	Orthogonal Least Squares [16]
MSV	Minimum Singular Value (proposed)
LRIP	Lower Restricted Isometry Property constant (adapted from [10])
SAS	Smallest Angle with selected Subspace (proposed)
ACD	Alternating Coordinate Descent [1]

Table 3: Greedy TLS algorithms structure.

Input: $\boldsymbol{A}$, $\boldsymbol{b}$, s
1. for $j = 1 : n$
    1.1. $\mathcal{I}_j = \{j\}$
    1.2. for $i = 2 : s$
        1.2.1. Find "best" column $\boldsymbol{a}_k$, $k \notin \mathcal{I}_j$
        1.2.2. Increase support: $\mathcal{I}_j \leftarrow \mathcal{I}_j \cup \{k\}$
    1.3. Compute $v_j = \sigma_{\min}([\boldsymbol{A}_{\mathcal{I}_j}\ \boldsymbol{b}])$
Output: support $\mathcal{I}_j$ that gives the smallest v_j

greedy algorithms based on heuristics that are specific to the TLS problem. The names and acronyms of all algorithms are listed in Table 2, for easier further reference. We adopt the general structure from Table 3 for our greedy algorithms. Note that all columns of $\boldsymbol{A}$ are tried for the first position in $\mathcal{I}$, but the other positions are filled using the standard greedy strategy. The typical greedy approach without the outer j loop gives very poor results, at least combined with the heuristics given below.

For the "best" column selection in 1.2.1 we have used three heuristics, of which the first two are known, but have been used in a different context in [10].

The first heuristic is based on the minimum singular value (MSV) choice: the new index is the $k \notin \mathcal{I}$ (from now on we omit the index j from $\mathcal{I}_j$) for which $\sigma_{\min}([\boldsymbol{A}_{\mathcal{I}\cup\{k\}}\ \boldsymbol{b}])$ is minimum. MSV is for sparse TLS what OLS is for sparse LS, namely the algorithm that locally optimizes the same criterion

whose overall optimization is sought (OLS selects the column which, added to the previous ones, produces the smallest residual).

The second heuristic is borrowed from [10], where the lower RIP constant defined by (5) is estimated. This problem is similar with (2) by the fact that a subset of columns of a matrix is sought, such that their smallest singular value is as small as possible. In the sparse TLS case, the matrix is $[\boldsymbol{A}\ \boldsymbol{b}]$ and $\boldsymbol{b}$ must always be one of the selected $s+1$ columns. The greedy strategy from [10] is hence directly applicable, starting with $\boldsymbol{b}$ and then searching for s columns of $\boldsymbol{A}$ in the attempt of solving (2); we name LRIP this variation of the algorithm. Note that the algorithm from [10] uses mainly the basic loop structure from Table 3, although some variations are also possible.

The third heuristic, named SAS, is our contribution and consists of choosing the column that makes the smallest angle with the subspace generated by the already selected columns and $\boldsymbol{b}$. This way, we attempt to indirectly and approximately minimize the smallest singular value of the resulting submatrix (since this submatrix is presumably closer to singularity than the others), but also to approximately minimize the angle between $\boldsymbol{b}$ and the subspace of the selected columns, which would be equivalent to the OLS heuristic of minimizing the current residual; thus, SAS is a hybrid LS-TLS heuristic. Denoting $\mathcal{T}_{\mathcal{I}} = \mathrm{Im}([\boldsymbol{b}\ \boldsymbol{A}_{\mathcal{I}}])$, the "best" column is found by

$$k = \mathrm{argmax}_{\ell \notin \mathcal{I}} \frac{\mathrm{proj}_{\mathcal{T}_{\mathcal{I}}} \boldsymbol{a}_\ell}{\|\boldsymbol{a}_\ell\|}, \tag{24}$$

where $\mathrm{proj}_{\mathcal{T}_{\mathcal{I}}} \boldsymbol{a}_\ell$ denotes the projection of column ℓ of $\boldsymbol{A}$ on $\mathcal{T}_{\mathcal{I}}$. This projection can be computed easily if a partial QR factorization ($\mathcal{J}$ is the complement of $\mathcal{I}$)

$$\boldsymbol{Q}^T [\boldsymbol{b}\ \boldsymbol{A}_{\mathcal{I}}\ \boldsymbol{A}_{\mathcal{J}}] = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{P} \\ 0 & \boldsymbol{S} \end{bmatrix} \tag{25}$$

is available at iteration i of step 1.2, where $\boldsymbol{R} \in \mathbb{R}^{i \times i}$, $i = |\mathcal{I}| + 1$, is upper triangular and $\boldsymbol{Q}$ is orthogonal. After the orthogonal transformation, the matrix $\boldsymbol{P}$ contains the projections of columns $\boldsymbol{a}_\ell$, $\ell \notin \mathcal{I}$, on $\mathcal{T}_{\mathcal{I}}$ (while $\boldsymbol{S}$ contains the components orthogonal on $\mathcal{T}_{\mathcal{I}}$). Hence, the "best" column is that for which $\|\boldsymbol{p}_\ell\| / \|\boldsymbol{a}_\ell\|$ is maximum.

The content of the main loop of SAS greedy algorithm for TLS (the j loop in Table 3) can be organized as a standard QR factorization with column pivoting [18] of the matrix $\boldsymbol{C} = [\boldsymbol{b}\ \boldsymbol{A}]$, as shown in Table 4. The factorization is stopped after $s+1$ steps and modified such that there is

Table 4: SAS main loop operations organized as a QR factorization with pivoting.

Input: $\boldsymbol{A}$, $\boldsymbol{b}$, s
1. $\boldsymbol{C} = [\boldsymbol{b}\ \boldsymbol{A}]$
2. Compute Householder reflector $\boldsymbol{U}_0$ such that $\boldsymbol{U}_0\boldsymbol{b} = \alpha[1\ 0\ \ldots\ 0]^T$
3. Update $\boldsymbol{C} \leftarrow \boldsymbol{U}_0\boldsymbol{C}$
4. Initialize projections norms vector $\boldsymbol{t} \in \mathbb{R}^n$ with zero
5. Initialize permutation vector $\boldsymbol{\pi} = 1 : n$
6. $\mathcal{I} \leftarrow \emptyset$
7. for $i = 1 : s$
   - 7.1. Update projection norms: $\boldsymbol{t}(i:n) \leftarrow \boldsymbol{t}(i:n) + |\boldsymbol{C}(i, i+1:n+1)|^2$
   - 7.2. Find "best" column: $k \leftarrow \arg\max_{\ell=i:n} \boldsymbol{t}(\ell)/\|\boldsymbol{a}_\ell\|^2$
   - 7.3. Permute columns $i+1$ and $k+1$ of $\boldsymbol{C}$. Interchange $\boldsymbol{\pi}(k) \leftrightarrow \boldsymbol{\pi}(i)$
   - 7.4. Compute Householder reflector $\boldsymbol{U}_i$ such that column $i+1$ of $\boldsymbol{U}_i\boldsymbol{C}$ is zero below the diagonal
   - 7.5. Update $\boldsymbol{C} \leftarrow \boldsymbol{U}_i\boldsymbol{C}$
   - 7.6. Increase support: $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{\pi}(i)\}$

Output: support $\mathcal{I}$

no pivoting for the first column. The computation is organized in place. The pivoting rule is (24), which can be efficiently implemented taking into account that the first i rows of the matrix (25) are no more modified after step i of the factorization. The vector $\boldsymbol{t}$ stores the norms of the columns of the projections $\boldsymbol{P}$ from (25), which are updated only with the new row that enters $\boldsymbol{P}$ at each iteration i. The vector $\boldsymbol{\pi}$ stores the column permutations; at the end, its first s elements are the indices from $\mathcal{I}$. (The algorithm was actually implemented without explicit permutations, which made it faster in Matlab.) The complexity of iteration i 7.1–7.6 is $O((m-i)(n-i))$ and is similar to the corresponding operations of the algorithm from [10], although with a higher constant hidden by the $O()$ notation.

4. Simulations

To compare the algorithms for sparse TLS, we generate random $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ and their noisy versions $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{b}}$ as described in Section 2.4, but

without imposing bounds on the absolute values of the elements of the generating solution $\boldsymbol{x}$. The data $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{b}}$ are fed to the sparse TLS solvers, which return solutions $\hat{\boldsymbol{x}}$ having support $\hat{\mathcal{I}}$. The greedy algorithms are stopped after s nonzero elements of the solution are computed, hence with $|\hat{\mathcal{I}}| = s$.

The algorithm from [1], named ACD (Alternating Coordinate Descent) further on, is run with 20 values of the trade-off constant λ, logarithmically spaced between 0.001 and 1. The best result is reported, as detailed below. Typically, the best λ gives $|\hat{\mathcal{I}}| > s$; we also tried choosing a λ for which $|\hat{\mathcal{I}}| = s$, but this gave clearly worse results. In the spirit of our two-step approach, we have also added to ACD a final processing step: the support of the computed solution is used for computing the optimal TLS solution with that support.

For each set of values of the simulation parameters m, n, s and ς, we have run 500 times each solver and report the mean value of the ℓ_2 relative error of the solution

$$e_2 = E\left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|/\|\boldsymbol{x}\|\right]. \tag{26}$$

For ACD, the averages are made for each λ and we report the results for the value of λ that gives the smallest e_2.

We implemented all the algorithms in Matlab, without special care to optimize their speed, so we do not give many details on the execution time. Even for small problem sizes like $m = 20$, MSV may take more than 1 sec for solving one instance of (2), while OMP needs 1-2 ms. In general, the increasing order of execution time is: OMP, OLS, LRIP, ACD, SAS, MSV. There is a large gap between OLS and LRIP, then between SAS and MSV.

We report results for $m \in \{20, 50, 100\}$ and $n/m = 3$; the results for $n/m \in \{2, 4\}$ are more or less similar; we consider only the case $m < n$ (overcomplete dictionaries), since it is the one of interest for applications. The values of the SNR are 10, 20, 30 and 40 dB.

We report detailed results only for OMP, SAS and ACD. OLS is almost always slightly better than OMP; we choose to comment only on OMP, since it has the merit of being the fastest algorithm. MSV and LRIP have a surprisingly poor performance; at low SNR, they are good in minimizing the TLS criterion $\sigma_{\min}([\hat{\boldsymbol{A}}_{\hat{\mathcal{I}}}\ \hat{\boldsymbol{b}}])$; however, in terms of quality of the approximation (26) of the generating solution, they compete with the other algorithms only for very low sparsity level, otherwise being far behind; the only reason for mentioning them in this text is that they are based on appealing heuristics already existing in the literature.

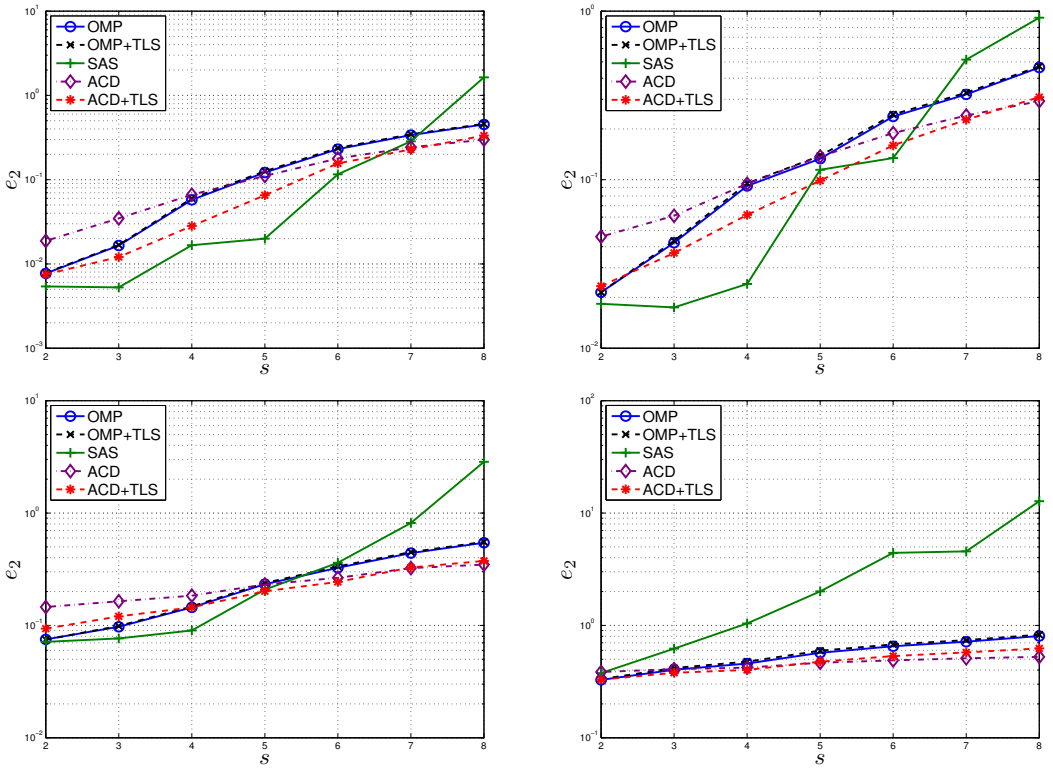

Figure 2: Mean solution error for $m = 20$, $n = 60$. The SNR is 40, 30 (upper row of figures), 20 and 10 dB (lower row).

The values of the criterion (26) are shown for $m = 20$, $n = 60$ (Figure 2), $m = 50$, $n = 150$ (Figure 3), $m = 100$, $n = 300$ (Figure 4), for different values of the sparsity s. The curves labeled OMP and ACD correspond to the results of the original algorithms (so OMP solves the sparse LS problem). The curves labeled OMP+TLS and ACD+TLS come from the two-steps strategy in which the original algorithm computes the support, then the TLS solution is found for that support; SAS is also based on the same strategy.

The two-step approach is definitely good for ACD; only for the lowest SNR and large values of s the algorithm ACD+TLS is slightly worse; in all the other cases it beats ACD, sometimes with a large margin. The differences between OMP and OMP+TLS are less visible, especially at high SNR; at low SNR, OMP+TLS is better for small sparsity level s, while OMP is slightly better for larger s. This fact may appear surprising, but has a sim-

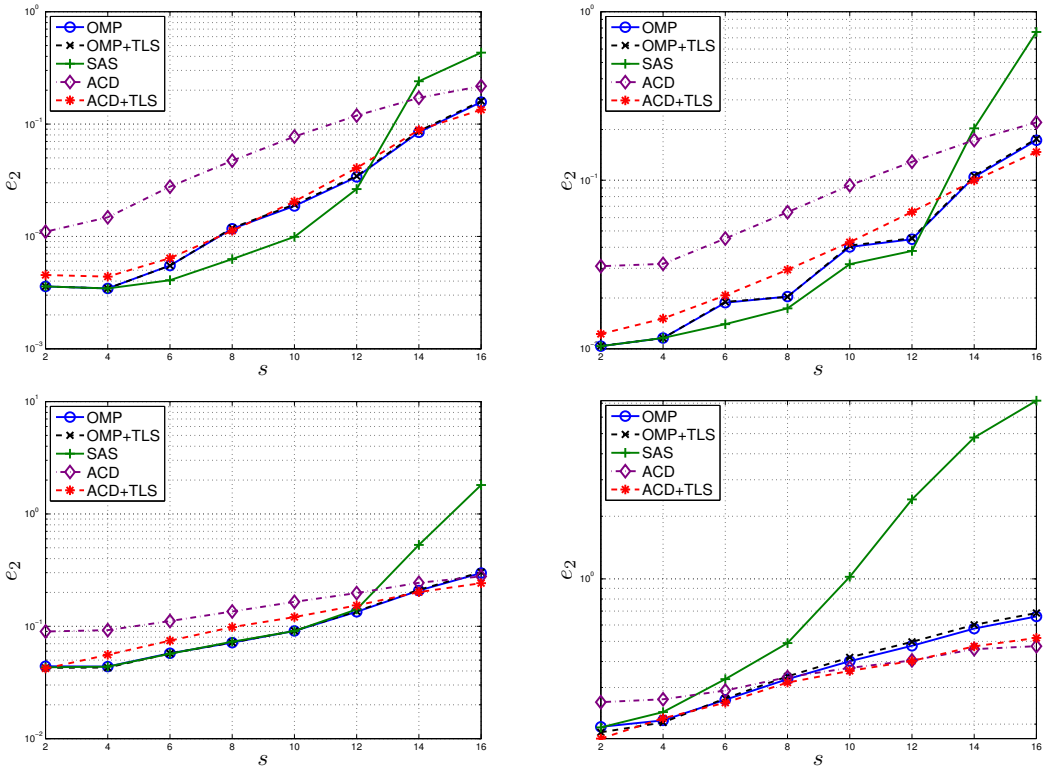

Figure 3: Mean solution error for $m = 50$, $n = 150$. The SNR is 40, 30 (upper row of figures), 20 and 10 dB (lower row).

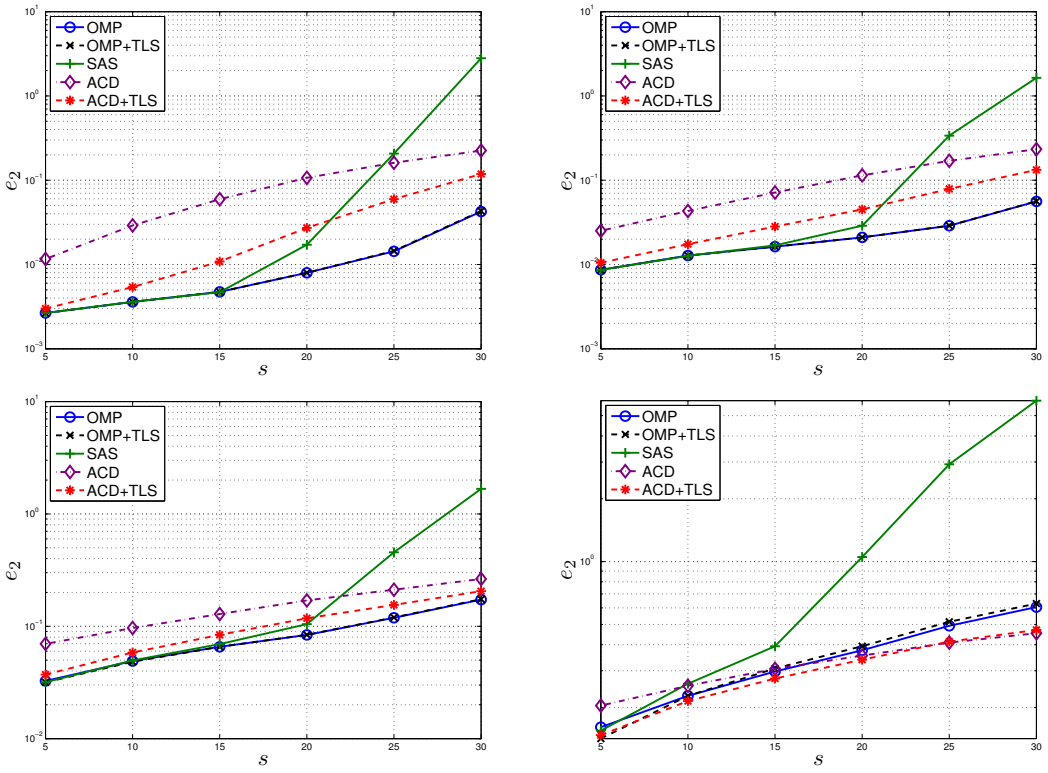

Figure 4: Mean solution error for $m = 100$, $n = 300$. The SNR is 40, 30 (upper row of figures), 20 and 10 dB (lower row).

ple explanation. When OMP finds the correct support, the TLS solution is usually closer to the generating solution (in average, not necessarily in each case); however, if the support is wrong, then the TLS solution appears to be usually farther than the LS one. A basic reason may be that TLS can be seen as a deregularized LS problem [8] and is more sensitive to perturbations. However, the above behavior occurs almost only to OMP; for the other studied algorithms, computing the LS solution instead of the TLS one on the final support rarely brings benefits; for example, for ACD this happens only for $m = 20$ and a few large values of s. As an empirical rule given by our inherently limited experiments with random matrices, we recommend to use the TLS solution on the support found by OMP for $s/m \leq 0.1$ (or even higher values for high SNR) and the LS solution otherwise.

Comparing now the methods, we note that SAS gives the best results for medium and high SNR and relatively small sparsity level, say, $s/m \leq 0.15$. However, its performance fades as the size of the problem grows; for $m = 100$, SAS is not better than OMP (by OMP we refer also to OMP+TLS). ACD+TLS is the best for low SNR (10dB) and also, occasionally, for large sparsity level, so it is somehow complementary to SAS in performance. Finally, OMP appears as the most robust method; it performs consistently well, being never far behind the best method; moreover, for the largest size we used, OMP is best for medium and high SNR and quite near the best at low SNR.

Since Theorem 4 and the numerical evidence from Section 2.4 suggest that, at high and (at a lesser extent) at medium SNR, the sparse LS and TLS optimal supports are the same in many cases, the good behavior of OMP compared with the other algorithms leads to the conclusion that the greedy LS heuristics work better than those inspired directly by TLS, especially as the dimension of the problem grows.

All these results should be taken with a caveat. For low SNR, the support of the best s-sparse LS or TLS solution may be different from the support of the generating solution: the noise affects not only the values of the solution, but it can also change its support. This was already remarked in Section 2.4, where we computed the optimal solutions by exhaustive search. So, the value of the criterion (26) may be worse for an algorithm because it is unable to find the optimal support when this is the original one, or it is able to find the optimal support when this is different from the original one. In our simulations, greedy algorithms were more prone to this kind of error, since they compute a support of size s. ACD is less sensitive, since it computes

Table 5: Support errors for $m = 50$, $n = 150$, SNR= 20 dB.

	s	2	4	6	8	10	12	14	16
Support misses	OMP	0.08	0.31	0.53	0.87	1.37	2.04	3.16	4.67
	SAS	0.09	0.32	0.54	0.83	1.37	2.25	3.77	5.65
	ACD	0.05	0.20	0.33	0.56	0.90	1.23	1.76	2.51
Wrong support	ACD	6.51	15.4	21.1	19.5	19.7	26.6	26.8	27.9

a larger support, which is more likely to include the correct one; however, this increases the errors by considering as nonzero some zero values of the solution. The different behavior of the algorithms in finding the true support is illustrated in Table 5, for the case $m = 50$, $n = 150$, SNR= 20 dB. The first three rows show the average number of positions of the true support that are missed by OMP, SAS and ACD, respectively. For OMP and SAS, the same numbers represent also the average number of wrong support positions that are considered as correct; for ACD, the number of wrong positions is shown on the last row of the table. In general, for OMP and SAS, the support errors are correlated with the mean solution errors: when a method is better for one type of error, it is usually better for the other also (or the difference is almost negligible). ACD has less true support misses, but a much larger number of wrong support positions.

5. Conclusions

We have presented theoretical (Theorem 4) and numerical evidence that the sparse LS and TLS solutions of a linear system may often have the same support. We have also shown that greedy algorithms can be an efficient approach for solving sparse TLS problems. In particular, orthogonal matching pursuit (OMP) can be used to recover the support of the TLS solution, after which the solution can be computed with standard TLS solvers based on the singular value decompositions. Since OMP is clearly the fastest algorithms among those presented here and in the previous works [1], [10] and also behaved well in our simulations, it can be recommended as the first choice solver for sparse TLS problems.

6. Acknowledgement

This work was supported by Tekes Finland Distinguished Professor (FiDiPro) Programme and by Romanian National Authority for Scientific Research, CNCS UEFISCDI, under Project PN-II-ID-PCE-2011-3-0400.

References

[1] H. Zhu, G. Leus, G. Giannakis, Sparsity-Cognizant Total Least-Squares for Perturbed Compressive Sampling, IEEE Trans. Signal Proc. 59 (5) (2011) 2002–2016.

[2] M. Herman, T. Strohmer, General Deviants: An Analysis of Perturbations in Compressed Sensing, IEEE J. Sel. Topics Signal Proc. 4 (2) (2010) 342–349.

[3] M. Herman, D. Needell, Mixed Operators in Compressed Sensing, in: 44th Conf. Information Sciences and Systems (CISS), 2010, pp. 1–6.

[4] J. Ding, L. Chen, Y. Gu, Performance Analysis of Orthogonal Matching Pursuit under General Perturbations, submitted to IEEE Trans. Signal Proc., available at http://arxiv.org/abs/1106.3373 (2011).

[5] D. Chae, P. Sadeghi, R. Kennedy, Effects of Basis-mismatch in Compressive Sampling of Continuous Sinusoidal Signals, in: 2nd Conf. Future Computer and Communication, Vol. 2, 2010, pp. 739–743.

[6] Y. Chi, L. Scharf, A. Pezeshki, A. Calderbank, Sensitivity to Basis Mismatch in Compressed Sensing, IEEE Trans. Signal Proc. 59 (5) (2011) 2182–2195.

[7] J. Parker, V. Cevher, P. Schniter, Compressive Sensing under Matrix Uncertainties: An Approximate Message Passing Approach, in: 45th Asilomar Conf. Signals Systems Computers, Pacific Grove, CA, 2011.

[8] I. Markovsky, S. Van Huffel, Overview of Total Least-Squares Methods, Signal Proc. 87 (10) (2007) 2283–2302.

[9] S. Van Huffel, J. Vandewalle, The Total Least Squares Problem: Computational Aspects and Analysis, SIAM, 1991.

[10] C. Dossal, G. Peyré, J. Fadili, A Numerical Exploration of Compressed Sampling Recovery, Lin. Alg. Appl. 432 (2010) 1663–1679.

[11] T. Tony Cai, L. Wang, Orthogonal matching pursuit for sparse signal recovery with noise, IEEE Trans. Information Theory 57 (7) (2011) 4680–4688.

[12] E. Candes, T. Tao, Decoding by Linear Programming, IEEE Trans. Info. Theory 51 (12) (2005) 4203–4215.

[13] C. Paige, Z. Strakoš, Bounds for the Least Squares Distance Using Scaled Total Least Squares, Numerische Mathematik 91 (2002) 93–115.

[14] M. Journée, Y. Nesterov, P. Richtárik, R. Sepulchre, Generalized Power Method for Sparse Principal Component Analysis, J. Machine Learning Res. 11 (2010) 517–553.

[15] J. Blanchard, C. Cartis, J. Tanner, Compressed Sensing: How Sharp is the Restricted Isometry Property?, SIAM Rev. 53 (1) (2009) 105–125.

[16] S. Chen, S. Billings, W. Luo, Orthogonal Least Squares Methods and Their Application to Non-Linear System Identification, Int. J. Control 50 (5) (1989) 1873–1896.

[17] M. Elad, Sparse and Redundant Representations: from Theory to Applications in Signal Processing, Springer, 2010.

[18] G. Golub, C. Van Loan, Matrix Computation, Johns Hopkins University Press, 1996.