HAL

archives-ouvertes.fr

Finding sparse solutions of systems of polynomial equations via group-sparsity optimization

Fabien Lauer, Henrik Ohlsson

Finding sparse solutions of systems of polynomial equations via group-sparsity optimization

F. Lauer[1] and H. Ohlsson[2,3]

[1] LORIA, Université de Lorraine, CNRS, Inria, France
[2] Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA
[3] Dept. of Electrical Engineering, Linköping University, Sweden

July 16, 2014

Abstract

The paper deals with the problem of finding sparse solutions to systems of polynomial equations possibly perturbed by noise. In particular, we show how these solutions can be recovered from group-sparse solutions of a derived system of linear equations. Then, two approaches are considered to find these group-sparse solutions. The first one is based on a convex relaxation resulting in a second-order cone programming formulation which can benefit from efficient reweighting techniques for sparsity enhancement. For this approach, sufficient conditions for the exact recovery of the sparsest solution to the polynomial system are derived in the noiseless setting, while stable recovery results are obtained for the noisy case. Though lacking a similar analysis, the second approach provides a more computationally efficient algorithm based on a greedy strategy adding the groups one-by-one. With respect to previous work, the proposed methods recover the sparsest solution in a very short computing time while remaining at least as accurate in terms of the probability of success. This probability is empirically analyzed to emphasize the relationship between the ability of the methods to solve the polynomial system and the sparsity of the solution.

1 Introduction

When faced with an underdetermined system of equations, one typically applies a regularization strategy in order to recover well-posedness. The choice of regularization depends on the particular application at hand and should be made to drive the solution towards desired properties. In the absence of precise goals, the most popular choice favors solutions with minimum ℓ_2-norm. However, an alternative becoming more and more popular is to search for sparse solutions (which often have non-minimal ℓ_2-norm). In the case of a system of linear equations, this alternative has been investigated in numerous works, see, e.g., [Bruckstein et al., 2009] for a review, and entails many applications of great importance, particularly for signal processing where it goes under the name of compressed sensing/sampling [Donoho, 2006, Candès, 2006]. Formally, finding the sparsest solutions of linear systems can be written as the minimization, under the constraints of the linear system, of the number of nonzero variables, which is a nonsmooth, nonconvex and NP-hard problem. Two main approaches can be distinguished to tackle such problems. The first one, known as Basis Pursuit (BP), relies on a convex relaxation based on the minimization of an ℓ_1-norm, while the second one applies a greedy strategy to add nonzero variables one-by-one. Many results on the convergence of these methods to the sparsest solution are available in the literature [Bruckstein et al., 2009, Eldar and Kutyniok, 2012, Foucart and Rauhut, 2013].

More recently, a few works introduced extensions of this problem to nonlinear equations. In particular, the first greedy approaches appeared in [Blumensath and Davies, 2008, Blumensath, 2013, Beck and Eldar, 2013, Ehler et al., 2014], while BP algorithms were developed in [Ohlsson et al., 2013b] to find sparse solutions of systems of quadratic equations and in

1

[Ohlsson et al., 2013a] for more general nonlinear equations. Formally, these problems can be formulated as

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 \tag{1}$$
$$\text{s.t. } y_i = f_i(\boldsymbol{x}), \quad i = 1, \ldots, N,$$

where $y_i \in \mathbb{R}$, $f_i : \mathbb{R}^n \to \mathbb{R}$ are nonlinear functions and $\|\boldsymbol{x}\|_0 = |\{j \in \{1, \ldots, n\} : x_j \neq 0\}|$ denotes the ℓ_0-pseudo-norm of $\boldsymbol{x}$, i.e, the number of nonzero components x_j.

Here, we focus on the case where the f_i's in (1) are polynomial functions of maximal degree d:

$$y_i = f_i(\boldsymbol{x}) = p_i^d(\boldsymbol{x}) = b_i + \sum_{k=1}^{M} a_{ik} \boldsymbol{x}^{\alpha_k}, \quad i = 1, \ldots, N, \tag{2}$$

where $\{\alpha_k\}_{k=1}^{M}$ is the set of $M = \sum_{q=1}^{d} \binom{n+q-1}{q}$ multi-indexes with $1 \leq |\alpha_k| = \sum_{j=1}^{n} (\alpha_k)_j \leq d$ and $\boldsymbol{x}^{\alpha_k} = \prod_{j=1}^{n} x_j^{(\alpha_k)_j}$ are the corresponding monomials. This includes the particular case considered in [Ohlsson et al., 2013b] with $d = 2$, while in [Ohlsson et al., 2013a] this setting is used to deal with the more general case via the Taylor expansions of the nonlinear functions f_i. Note that the formulation in (1)–(2) also entails cases outside of the quasi-linear setting considered in [Ehler et al., 2014].

The present paper proposes a new BP approach to solve (1)–(2), which, in comparison with the previous works [Ohlsson et al., 2013b, Ohlsson et al., 2013a], is more simple and amounts to solving an easier optimization problem. More precisely, the method proposed in Sect. 2.1 relies on a simple change of variable sufficient to result in an efficient algorithm implemented as a classical ℓ_1-minimization problem. However, the structure of the polynomials is discarded and the solution may not satisfy the original polynomial constraints. Note that this change of variable is also found in [Ohlsson et al., 2013a], though with constraints handled in a more complex manner, and closely related to the lifting technique of [Ohlsson et al., 2013b] for quadratic BP and the one of [Vidal et al., 2005] for subspace clustering.

The method in [Ohlsson et al., 2013b] enforces the structure of the quadratic polynomials via rank constraints, which leads to optimization problems with additional levels of relaxation and the introduction of a parameter tuning the trade-off between the minimization of the ℓ_1-norm on the one hand and the satisfaction of the rank constraint on the other hand. In [Ohlsson et al., 2013a], the structure of higher-degree polynomials is enforced by a set of quadratic constraints on the monomials[1], which are then relaxed as in [Ohlsson et al., 2013b].

Instead, the method proposed in Sect. 2.2 implements the structural knowledge via group-sparsity. This results in an ℓ_1/ℓ_2-minimization, i.e., a second-order cone program (SOCP) which can be solved more efficiently. In addition, this SOCP formulation is easily extended in Sect. 2.4 to benefit from reweighting techniques for improving the sparsity of the solution.

The conditions for exact recovery of the proposed BP methods are analyzed in Sect. 2.3. In particular, we show that though the simple ℓ_1-minimization does not include structural constraints, exact recovery occurs for sufficiently sparse cases. A similar condition is proved for the ℓ_1/ℓ_2-minimization based on group-sparsity.

The greedy approach is discussed in Sect. 3, where two variants are proposed: an exact algorithm for solving the group-sparsity optimization problem in small-scale cases and an approximate one which remains efficient in higher-dimensional settings. Previous greedy approaches [Blumensath, 2013, Beck and Eldar, 2013] considered the problem in its sparsity-constrained form, where the sum of squared errors over the equations (2) is minimized subject to $\|\boldsymbol{x}\|_0 \leq s$ for an *a priori* fixed s. The iterative hard thresholding of [Blumensath, 2013] is a gradient descent algorithm with an additional projection onto the feasible set at each iteration via a simple thresholding operation. In [Beck and Eldar, 2013], this is interpreted as a fixed point iteration enforcing a necessary (but not sufficient) optimality condition, which requires a well-chosen step-size to converge

---

[1]For example, consider the monomials $u_1 = x_1$, $u_2 = x_2$, $u_3 = x_1 x_2$, $u_4 = x_1^2 x_2$, then the structure of u_3 and u_4 is enforced by $u_3 = u_1 u_2$ and $u_4 = u_1 u_3$. But note that since these constraints are relaxed in the final formulation, the estimation can yield $u_3 \neq u_1 u_2$, which then recursively implies that all monomial constraints involving u_3 are meaningless.

satisfactorily. In particular, the step-size must be an upper bound on the Lipschitz constant of the gradient of the objective function, which however is not (globally) Lipschitz continuous for polynomials in (2) with degree $d > 1$. The sparse-simplex algorithm of [Beck and Eldar, 2013] is a coordinate descent method which enjoys similar but less restrictive convergence properties while being parameter-free. However, for polynomial equations as in (2), each iteration requires solving several one-dimensional minimizations of a polynomial of degree $2d$, which becomes difficult for $d > 2$. On the contrary, the proposed greedy algorithms remain simple thanks to the group-sparse and linearized formulation of the problem: each iteration requires only solving least-squares problems.

Extensions of the methods are discussed in Sections 4 and 5. In particular, Section 4 deals with the issue of purely nonlinear polynomials, for which the solution to (1)–(2) cannot be estimated as directly as for polynomials involving linear monomials, but which arise in important applications such as phase retrieval [Kohler and Mandel, 1973, Gonsalves, 1976, Gerchberg and Saxton, 1972, Fienup, 1982, Marchesini, 2007, Candès et al., 2013b, Candès et al., 2013a]. Then, the case where the equations (2) are perturbed by noise is considered in Sect. 5. In particular, the analysis provides stable recovery results for polynomial BP denoising in the flavor of the one obtained in [Donoho et al., 2006] for linear BP denoising.

Finally, numerical experiments in Sect. 6 show the efficiency of the proposed methods and extensions. Results are in line with the ones found in classical sparse optimization with linear constraints. In particular, all methods can recover the sparsest solution in sufficiently sparse cases and the greedy approach is the fastest while the BP methods based on convex relaxations benefit from a slightly higher probability of success.

2 Polynomial basis pursuit

This section develops two basis pursuit approaches to find sparse solutions of systems of polynomial equations via the minimization of an ℓ_1-norm for the first one and of a mixed ℓ_1/ℓ_2-norm for the second one. The methods are developed under the following assumption, which will be relaxed in Sect. 4.

Assumption 1. *For all $j \in \{1, \ldots, n\}$, $\exists i \in \{1, \ldots, N\}$ such that $a_{ik} \neq 0$ for k determined such that $(\alpha_k)_l = 1$ for $l = j$ and $(\alpha_k)_l = 0$ for $l \in \{1, \ldots, n\} \setminus \{j\}$.*

In particular, Assumption 1 ensures that the polynomials include a linear part, or more precisely, that for any variable x_j, $j = 1, \ldots, n$, the monomial x_j has a nonzero coefficient in at least one of the polynomials.

2.1 Polynomial basis pursuit via ℓ_1-minimization

We start by rewriting the constraints in (1)–(2) as

$$y_i = f_i(\boldsymbol{x}) = b_i + \sum_{k=1}^{M} a_{ik} \boldsymbol{x}^{\alpha_k} = \boldsymbol{a}_i^T \phi(\boldsymbol{x}) + b_i, \tag{3}$$

where $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iM}]^T$ and $\phi : \mathbb{R}^n \to \mathbb{R}^M$ is a mapping that computes all the monomials of degree q, $1 \leq q \leq d$, with n variables. Note that $\boldsymbol{x}$ is embedded in $\phi(\boldsymbol{x})$ as the components corresponding to the n multi-indexes $\alpha_{k(j)}$, $j = 1, \ldots, n$, such that $(\alpha_{k(j)})_l = 1$ for $l = j$ and 0 for $l \neq j$. Assuming that these are the first components of $\phi(\boldsymbol{x})$, i.e., $k(j) = j$, we also define the (linear) inverse mapping $\phi^{-1} : \mathbb{R}^M \to \mathbb{R}^n$, such that $\phi^{-1}(\phi(\boldsymbol{x})) = \boldsymbol{x}$, as

$$\phi^{-1}(\boldsymbol{\phi}) = \begin{bmatrix} \boldsymbol{I}_n & 0 \end{bmatrix} \boldsymbol{\phi},$$

where $\boldsymbol{I}_n$ is the n-by-n identity matrix.

The high-dimensional lifting by ϕ allows us to recast (1)–(2) as a standard (i.e., linear) problem in sparse optimization:

$$\min_{\boldsymbol{\phi} \in \mathbb{R}^M} \|\boldsymbol{\phi}\|_0 \tag{4}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b},$$

3

where $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]^T$, $\boldsymbol{b} = [b_1, \ldots, b_N]^T$ and $\phi(\boldsymbol{x})$ is replaced by an unstructured vector $\boldsymbol{\phi}$. This constitutes the first level of relaxation in the proposed approach, where the components of $\boldsymbol{\phi}$ are not constrained to be interdependent monomials. While this yields a rather crude approximation, it will serve as the basis for the refined approach of Sect. 2.2, where additional structure will be imposed on $\boldsymbol{\phi}$.

The second level of relaxation comes from the BP approach, in which problems such as (4) are typically solved via the convex relaxation

$$\hat{\boldsymbol{\phi}} = \arg \min_{\boldsymbol{\phi} \in \mathbb{R}^M} \ \|\boldsymbol{W}\boldsymbol{\phi}\|_1 \tag{5}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b},$$

where $\boldsymbol{W} = diag\,(\|\boldsymbol{A}_1\|_2, \ldots, \|\boldsymbol{A}_M\|_2)$, with $\boldsymbol{A}_k$ the kth column of $\boldsymbol{A}$, is a diagonal matrix of precompensating weights. Then, for polynomial BP, an estimate of the solution to (1)–(2) can be easily computed under Assumption 1 as $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}})$.

Problem (5) can be formulated as a linear program and solved by generic interior-point algorithms in polynomial time, while it can also be solved by more efficient and dedicated algorithms, see, e.g., [Foucart and Rauhut, 2013, Chapter 15]. However, the complexity with respect to n remains exponential because of the dimension M which scales exponentially with n.

The literature on basis pursuit and compressed sensing [Bruckstein et al., 2009, Foucart and Rauhut, 2013] tells us that the sparser the solution to (4) is, the more likely the convex relaxation (5) is to yield its recovery. Here, by construction we know that there is at least a very sparse vector $\phi(\boldsymbol{x}_0)$ satisfying the constraints: with $\boldsymbol{x}_0$ the sparse solution to (1)–(2), the sparsity level $\|\phi(\boldsymbol{x}_0)\|_0/M$ is better than $\|\boldsymbol{x}_0\|_0/n$, as stated by Proposition 1 below. Note that a better bound implying an increased level of sparsity depending on d for very sparse cases is derived in Appendix A.

Proposition 1. *Let the mapping $\phi : \mathbb{R}^n \to \mathbb{R}^M$ be defined as above. Then, the vector $\phi(\boldsymbol{x})$ is at least as sparse as the vector $\boldsymbol{x}$ in the sense that the inequality*

$$\frac{\|\phi(\boldsymbol{x})\|_0}{M} \leq \frac{\|\boldsymbol{x}\|_0}{n}$$

holds for all $\boldsymbol{x} \in \mathbb{R}^n$.

Proposition 1 implies that if (1)–(2) has a sparse solution then so does (4). To prove Proposition 1, we first need the following lemma.

Lemma 1. *For all triplet $(a, b, c) \in (\mathbb{N}^*)^3$ such that $a \geq b$, the inequality*

$$\frac{1}{a}\binom{a+c-1}{c} \geq \frac{1}{b}\binom{b+c-1}{c}$$

holds.

Proof. On the one hand, we have

$$\frac{1}{a}\binom{a+c-1}{c} = \frac{(a+c-1)!}{a\,c!\,(a-1)!} = \frac{1}{a\,c!}\prod_{i=0}^{c-1}(a+i) = \frac{1}{c!}\prod_{i=1}^{c-1}(a+i)$$

and a similar expression with b instead of a. On the other hand, with $a \geq b$, we have

$$\forall i \in \mathbb{N}, \ a+i \geq b+i \quad \Rightarrow \quad \frac{1}{c!}\prod_{i=1}^{c-1}(a+i) \geq \frac{1}{c!}\prod_{i=1}^{c-1}(b+i)$$

which then yields the sought statement.

$\square$

We now give the proof of Proposition 1.

Proof. By construction, the number of nonzeros in $\phi(\boldsymbol{x})$ is equal to the sum over q, $1 \leq q \leq d$, of the number of monomials of degree q in $\|\boldsymbol{x}\|_0$ variables. This yields

$$\frac{\|\phi(\boldsymbol{x})\|_0}{M} = \frac{\sum_{q=1}^{d} \binom{\|\boldsymbol{x}_0\|_0 + q - 1}{q}}{\sum_{q=1}^{d} \binom{n + q - 1}{q}} = \frac{\|\boldsymbol{x}\|_0}{n} \frac{\frac{1}{\|\boldsymbol{x}\|_0} \sum_{q=1}^{d} \binom{\|\boldsymbol{x}_0\|_0 + q - 1}{q}}{\frac{1}{n} \sum_{q=1}^{d} \binom{n + q - 1}{q}}$$

Since $n \geq \|\boldsymbol{x}\|_0$, Lemma 1 gives the bound

$$\frac{1}{\|\boldsymbol{x}\|_0} \sum_{q=1}^{d} \binom{\|\boldsymbol{x}_0\|_0 + q - 1}{q} \leq \frac{1}{n} \sum_{q=1}^{d} \binom{n + q - 1}{q}$$

from which the statement follows.

$\square$

2.2 Polynomial basis pursuit via ℓ_1/ℓ_2-minimization (group sparsity)

The approach proposed above relies on a rather crude approximation. Thus, the polynomial equations are not guaranteed to be satisfied by the solution $\hat{\boldsymbol{x}}$, due to the factorization of the polynomial that may not hold for $\hat{\phi}$. In other words, the linearization in (3) together with the direct optimization of ϕ discard the desired structure for ϕ. In practice, the solution $\hat{\boldsymbol{x}}$ can be checked a posteriori with $y_i \stackrel{?}{=} p_i^d(\hat{\boldsymbol{x}})$, $i = 1, \ldots, N$. But in order to increase the probability of obtaining a satisfactory solution, i.e., one which satisfies the original polynomial constraints, we must embed the structure of the polynomial in the problem formulation.

Let us define the index sets

$$I_j = \{k \in \{1, \ldots, M\} : (\alpha_k)_j \neq 0\}, \quad j = 1, \ldots, n. \tag{6}$$

Then, the structural information we aim at embedding is given by the following implication:

$$\forall j \in \{1, \ldots, n\}, \ \forall k \in I_j, \quad \begin{cases} x_j = 0 \ \Rightarrow \ (\phi(\boldsymbol{x}))_k = 0, \\ (\phi(\boldsymbol{x}))_k \neq 0 \ \Rightarrow \ x_j \neq 0, \end{cases} \tag{7}$$

which formalizes the fact that whenever a variable is zero, all monomials involving this variable must be zero. Such a structure can be favored via group-sparsity optimization, as detailed next.

Let us define the set of mappings $\varphi_j : \mathbb{R}^n \to \mathbb{R}^m$, each computing the subset of components of ϕ involving the variable x_j (see Appendix C for the precise value of $m = |I_j|$). Note that these mappings are nonlinear in $\boldsymbol{x}$ but linear in $\phi(\boldsymbol{x})$: $\varphi_j(\boldsymbol{x}) = \boldsymbol{B}_j \phi(\boldsymbol{x})$, where $\boldsymbol{B}_j$ is an m-by-M binary matrix filled with zeros except for a 1 on each row at the kth column for all $k \in I_j$. Further define the ℓ_0-pseudo-norm of a vector-valued sequence as the number of nonzero vectors in the sequence:

$$\|\{\boldsymbol{u}_j\}_{j=1}^{n}\|_0 = |\{j \in \{1, \ldots, n\} : \boldsymbol{u}_j \neq \boldsymbol{0}\}|.$$

We call ϕ a group-sparse vector if $\|\{\boldsymbol{B}_j \phi\}_{j=1}^{n}\|_0$ is small. By construction and due to (7), a sparse $\boldsymbol{x}$ leads to a group-sparse $\phi(\boldsymbol{x})$ with

$$\|\{\boldsymbol{B}_j \phi(\boldsymbol{x})\}_{j=1}^{n}\|_0 = \|\boldsymbol{x}\|_0.$$

Therefore, in order to solve (1)–(2) we search for a group-sparse solution to the linear system $\boldsymbol{A}\phi = \boldsymbol{y} - \boldsymbol{b}$.

Such group-sparse solutions can be found by solving

$$\min_{\phi \in \mathbb{R}^M} \|\{\boldsymbol{W}_j \phi\}_{j=1}^{n}\|_0 \tag{8}$$

$$\text{s.t. } \boldsymbol{A}\phi = \boldsymbol{y} - \boldsymbol{b},$$

where we replaced each $\boldsymbol{B}_j$ by a weight matrix $\boldsymbol{W}_j = \boldsymbol{B}_j \boldsymbol{W}$, with $\boldsymbol{W} = diag\left(\|\boldsymbol{A}_1\|_2, \ldots, \|\boldsymbol{A}_M\|_2\right)$, without effect on the ℓ_0-pseudo-norm. The reason for introducing these precompensating weights will become clear in the analysis of Sect. 2.3.

Let us define the ℓ_p/ℓ_q-norm of a vector-valued sequence as

$$\|\{\boldsymbol{u}_j\}_{j=1}^n\|_{p,q} = \left(\sum_{j=1}^n \|\boldsymbol{u}_j\|_q^p\right)^{\frac{1}{p}}.$$

Problem (8) can be relaxed by replacing the ℓ_0-pseudo-norm with an ℓ_p/ℓ_q-norm. In this paper we only consider the case $p = 1$ and $q = 2$, i.e.,

$$\|\{\boldsymbol{W}_j\boldsymbol{\phi}\}_{j=1}^n\|_{1,2} = \left\|\begin{pmatrix}\|\boldsymbol{W}_1\boldsymbol{\phi}\|_2\\ \vdots \\ \|\boldsymbol{W}_n\boldsymbol{\phi}\|_2\end{pmatrix}\right\|_1,$$

but other norms could be used, such as the ℓ_1/ℓ_∞-norm. This yields the convex relaxation[2]

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^M} \sum_{j=1}^n \|\boldsymbol{W}_j\boldsymbol{\phi}\|_2 \qquad (9)$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b},$$

which is easily reformulated as a Second Order Cone Program (SOCP) that can be solved by generic software such as CVX [Grant and Boyd, 2013, Grant and Boyd, 2008] or MOSEK [Andersen and Andersen, 2000] (more efficient dedicated solvers can also be found, e.g., [Deng et al., 2011]). However, as for the ℓ_1-minimization (5), the complexity remains exponential with respect to the number n of base variables via the dimension M.

Adding structure via constraints. All components of the mapping $\phi(\boldsymbol{x})$ involving only even degrees of base variables x_j are nonnegative. These structural constraints can help to drive the solution towards one that correctly factorizes and corresponds to a solution of the polynomial equations. This is obtained by solving

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^M} \sum_{j=1}^n \|\boldsymbol{W}_j\boldsymbol{\phi}\|_2 \qquad (10)$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b}$$
$$\phi_k \geq 0, \ \forall k \text{ such that } (\alpha_k)_j \text{ is even for all } j.$$

This optimization problem has the same complexity as (9) since we simply added nonnegativity constraints to some variables.

Other forms of prior knowledge can be easily introduced in (10). For instance, if (tight) box constraints on the base variables are available, then lower and upper bounds on all monomials can be derived. Finally, note that these structural constraints can also be added to the ℓ_1-minimization (5).

2.3 Analysis

The following derives conditions of exact recovery of the sparse solution to the system of polynomial equations via various convex relaxations. These conditions are based on the mutual coherence of the matrix $\boldsymbol{A}$ as defined e.g. in [Donoho and Huo, 2001, Bruckstein et al., 2009].

Definition 1. *The* mutual coherence *of a matrix $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_M]$ is*

$$\mu(\boldsymbol{A}) = \max_{1\leq i<j\leq M} \frac{|\boldsymbol{A}_i^T\boldsymbol{A}_j|}{\|\boldsymbol{A}_i\|_2\|\boldsymbol{A}_j\|_2}.$$

---

[2]Note that since all the groups have the same number of variables, they need not be weighted by a function of the number of variables in each group.

In order for the mutual coherence of $\boldsymbol{A}$ to be defined, we focus on the case where the following assumption holds.

Assumption 2. *All columns $\boldsymbol{A}_k$ of the matrix $\boldsymbol{A}$ are nonzero, i.e., $\boldsymbol{A}_k \neq \boldsymbol{0}$, $k = 1, \ldots, M$, or, equivalently, for all $k \in \{1, \ldots, M\}$, $\exists i \in \{1, \ldots, N\}$ such that the corresponding polynomial coefficient $a_{ik} \neq 0$.*

Assumption 2 is slightly more restrictive than Assumption 1, which only constrains the first n columns of $\boldsymbol{A}$. If Assumption 2 does not hold, the following analysis can be reproduced under Assumption 1 by considering the submatrix $\tilde{\boldsymbol{A}} \in \mathbb{R}^{N \times \tilde{M}}$ containing the $\tilde{M}$ nonzero columns of $\boldsymbol{A}$ and a similarly truncated mapping $\phi : \mathbb{R}^n \to \mathbb{R}^{\tilde{M}}$. Adjustments then need to be made where numbers of columns are used, i.e., by substituting $\tilde{M} \leq M$ and $\tilde{m}_j \leq m$ for M and m. However, for the case where both Assumptions 1 and 2 do not hold, i.e., of zero columns corresponding to base variables, $\boldsymbol{A}_k = \boldsymbol{0}$, $k \in \{1, \ldots, n\}$, $\hat{\boldsymbol{x}}$ cannot be obtained by the inverse mapping ϕ^{-1}. This particular case will be discussed in Sect. 4.

Note that whenever a condition requires the mutual coherence $\mu(\boldsymbol{A})$ to be defined, Assumption 2 implicitly holds. In such cases, Assumption 2 will not be explicitly stated in the theorems below.

2.3.1 ℓ_1-minimization method

The result below characterizes a case where the simple ℓ_1-minimization method is sufficient to solve the sparse optimization problem (1)–(2).

Theorem 1. *Let $\boldsymbol{x}_0$ denote the unique solution to (1)–(2). If the inequality*

$$\|\boldsymbol{x}_0\|_0 < \frac{n}{2M} \left(1 + \frac{1}{\mu(\boldsymbol{A})} \right) \tag{11}$$

holds, then the solution $\hat{\boldsymbol{\phi}}$ to (5) is unique and equal to $\phi(\boldsymbol{x}_0)$, thus providing $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}}) = \boldsymbol{x}_0$.

Proof. Assume (1)–(2) has a unique solution $\boldsymbol{x}_0$. Then, $\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b}$ has a solution $\boldsymbol{\phi}_0 = \phi(\boldsymbol{x}_0)$ with a sparsity bounded by Proposition 1 as

$$\|\boldsymbol{\phi}_0\|_0 \leq \frac{M}{n} \|\boldsymbol{x}_0\|_0.$$

But, according to Theorem 7 in [Bruckstein et al., 2009], if

$$\|\boldsymbol{\phi}_0\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\boldsymbol{A})} \right)$$

then, on the one hand, $\boldsymbol{\phi}_0$ is the sparsest solution to $\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b}$, and on the other hand, it is also the unique solution to (5). Thus, if (11) holds, $\boldsymbol{\phi}_0$ is the unique solution to both (4) and (5), i.e., $\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}_0$. Since $\hat{\boldsymbol{x}}$ is given by the first components of $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{x}_0$ by the ones of $\boldsymbol{\phi}_0$, this completes the proof.

$\square$

Other less conservative conditions can be similarly obtained by considering the exact value of $\|\boldsymbol{\phi}_0\|_0$ or tighter bounds (see Appendix B), but these do not take the form of a simple inequality on $\|\boldsymbol{x}_0\|_0$. Another condition for very sparse cases (with $\|\boldsymbol{x}_0\|_0 \leq n/d - 1$) can be similarly obtained by using Proposition 2 in Appendix A instead of Proposition 1.

2.3.2 ℓ_1/ℓ_2-minimization method

The first result below shows that the group-sparse problem (8) can be used as a proxy for the polynomial problem (1)–(2).

Theorem 2. *If the solution $\boldsymbol{\phi}^*$ to (8) is unique and yields $\boldsymbol{x}^* = \phi^{-1}(\boldsymbol{\phi}^*)$ such that $\boldsymbol{x}^*$ is a solution to the system of polynomial equations (2), then $\boldsymbol{x}^*$ is the sparsest solution to the system of polynomial equations, i.e., the unique solution to (1)–(2).*

Proof. Assume there is an $\boldsymbol{x}_0 \neq \boldsymbol{x}^*$ solution to the polynomial system (2) and at least as sparse as $\boldsymbol{x}^*$. Then

$$\boldsymbol{A}\phi(\boldsymbol{x}_0) = \boldsymbol{y} - \boldsymbol{b}$$

and

$$\|\{\boldsymbol{W}_j\phi(\boldsymbol{x}_0)\}_{j=1}^n\|_0 = \|\boldsymbol{x}_0\|_0 \leq \|\boldsymbol{x}^*\|_0 \leq \|\{\boldsymbol{W}_j\phi^*\}_{j=1}^n\|_0,$$

which contradicts the fact that ϕ^* is the unique solution to (8) unless $\phi(\boldsymbol{x}_0) = \phi^*$ and $\boldsymbol{x}_0 = \boldsymbol{x}^*$. $\qquad\square$

The next result provides a condition on the sparsity of the solution to (1)–(2) under which it can be recovered by solving the convex problem (9) or its variant with nonnegativity constraints (10). This result requires the following lemma.

Lemma 2. *Let $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_M]$ be an $N \times M$ matrix with mutual coherence $\mu(\boldsymbol{A})$ as defined in Definition 1. Let $\boldsymbol{W}$ be the $M \times M$-diagonal matrix of entries $w_i = \|\boldsymbol{A}_i\|_2$. Then, for all $\boldsymbol{\delta} \in Ker(\boldsymbol{A})$ and $i \in \{1, \ldots, M\}$, the bound*

$$w_i^2 \delta_i^2 \leq \frac{\mu^2(\boldsymbol{A})}{1 + \mu^2(\boldsymbol{A})} \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \qquad (12)$$

holds.

Proof. For all $\boldsymbol{\delta} \in Ker(\boldsymbol{A})$, we have $\boldsymbol{A}\boldsymbol{\delta} = 0 \Rightarrow \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\delta} = 0$, which further implies $\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1}\boldsymbol{W}\boldsymbol{\delta} = 0$ and

$$-\boldsymbol{W}\boldsymbol{\delta} = \left(\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I}\right)\boldsymbol{W}\boldsymbol{\delta}.$$

Then, we have

$$w_i^2\delta_i^2 = (\boldsymbol{W}\boldsymbol{\delta})_i^2 = \left((\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I})\boldsymbol{W}\boldsymbol{\delta}\right)_i^2$$

$$= \left(\sum_{j=1}^M \left(\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I}\right)_{i,j}(\boldsymbol{W}\boldsymbol{\delta})_j\right)^2$$

$$\leq \sum_{j=1}^M \left(\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I}\right)_{i,j}^2 (\boldsymbol{W}\boldsymbol{\delta})_j^2.$$

Note that $\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1}$ is a normalized matrix with ones on the diagonal and off diagonal (i,j)-entries equal to $\frac{|\boldsymbol{A}_i^T\boldsymbol{A}_j|}{\|\boldsymbol{A}_i\|_2\|\boldsymbol{A}_j\|_2}$ and bounded by $\mu(\boldsymbol{A})$ via Definition 1. Thus, we obtain

$$w_i^2\delta_i^2 \leq \mu^2(\boldsymbol{A})\sum_{j\neq i} w_j^2\delta_j^2$$

and, by adding $\mu^2(\boldsymbol{A})w_i^2\delta_i^2$ to both sides, the bound

$$w_i^2\delta_i^2 \leq \frac{\mu^2(\boldsymbol{A})}{1 + \mu^2(\boldsymbol{A})}\sum_{j=1}^M w_j^2\delta_j^2,$$

which is precisely (12). $\qquad\square$

Theorem 3. *Let $\boldsymbol{x}_0$ be a solution to the polynomial equations (2) and $\phi_0 = \phi(\boldsymbol{x}_0)$. If the condition*

$$\|\boldsymbol{x}_0\|_0 < \frac{1}{2\sqrt{m}}\sqrt{1 + \frac{1}{\mu^2(\boldsymbol{A})}},$$

where m is the number of components of ϕ associated to a variable, holds, then ϕ_0 is the unique solution to both (9) and (10).

Proof. The vector $\boldsymbol{\phi}_0$ is the unique solution to (9) if the inequality

$$\sum_{j=1}^{n} \|\boldsymbol{W}_j(\boldsymbol{\phi}_0 + \boldsymbol{\delta})\|_2 > \sum_{j=1}^{n} \|\boldsymbol{W}_j\boldsymbol{\phi}_0\|_2$$

holds for all $\boldsymbol{\delta} \neq \boldsymbol{0}$ satisfying $\boldsymbol{A}\boldsymbol{\delta} = 0$. The inequality above can be rewritten as

$$\sum_{j \in I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 + \sum_{j \notin I_0} \|\boldsymbol{W}_j(\boldsymbol{\phi}_0 + \boldsymbol{\delta})\|_2 - \|\boldsymbol{W}_j\boldsymbol{\phi}_0\|_2 > 0,$$

where $I_0 = \{j \in \{1, \ldots, n\} : \boldsymbol{W}_j\boldsymbol{\phi}_0 = \boldsymbol{0}\}$. By the triangle inequality, $\|\boldsymbol{u} + \boldsymbol{v}\|_2 - \|\boldsymbol{u}\|_2 \geq -\|\boldsymbol{v}\|_2$, this condition is met if

$$\sum_{j \in I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 - \sum_{j \notin I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 > 0$$

or

$$\sum_{j=1}^{n} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 - 2\sum_{j \notin I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 > 0. \tag{13}$$

By defining G_j as the set of indexes corresponding to nonzero columns of $\boldsymbol{W}_j$, Lemma 2 yields

$$\|\boldsymbol{W}_j\boldsymbol{\delta}\|_2^2 = \sum_{i \in G_j} w_i^2 \delta_i^2 \leq m_j \frac{\mu^2(\boldsymbol{A})}{1 + \mu^2(\boldsymbol{A})} \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2,$$

where $m_j = m$ is the number of components of ϕ associated to a variable x_j. Due to the fact that $\bigcup_{k \in \{1, \ldots, n\}} G_k = \{1, \ldots, M\}$, we also have

$$\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 = \sum_{i=1}^{M} w_i^2 \delta_i^2 \leq \sum_{k=1}^{n} \sum_{i \in G_k} w_i^2 \delta_i^2 = \sum_{k=1}^{n} \|\boldsymbol{W}_k\boldsymbol{\delta}\|_2^2 \leq \left(\sum_{k=1}^{n} \|\boldsymbol{W}_k\boldsymbol{\delta}\|_2\right)^2,$$

which then leads to

$$\|\boldsymbol{W}_j\boldsymbol{\delta}\|_2^2 \leq m \frac{\mu^2(\boldsymbol{A})}{1 + \mu^2(\boldsymbol{A})} \left(\sum_{k=1}^{n} \|\boldsymbol{W}_k\boldsymbol{\delta}\|_2\right)^2.$$

Introducing this result in (13) gives the condition

$$\sum_{j=1}^{n} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 - 2(n - |I_0|) \frac{\mu(\boldsymbol{A})\sqrt{m}}{\sqrt{1 + \mu^2(\boldsymbol{A})}} \sum_{k=1}^{n} \|\boldsymbol{W}_k\boldsymbol{\delta}\|_2 > 0.$$

Finally, given that $|I_0| = n - \|\{\boldsymbol{W}_j\boldsymbol{\phi}_0\}_{j=1}^{n}\|_0 = n - \|\boldsymbol{x}_0\|_0$, this yields

$$\sum_{j=1}^{n} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 - 2\|\boldsymbol{x}_0\|_0 \frac{\mu(\boldsymbol{A})\sqrt{m}}{\sqrt{1 + \mu^2(\boldsymbol{A})}} \sum_{k=1}^{n} \|\boldsymbol{W}_k\boldsymbol{\delta}\|_2 > 0. \tag{14}$$

or, after dividing by $\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^{n}\|_{1,2}$ and rearranging the terms,

$$\|\boldsymbol{x}_0\|_0 < \frac{\sqrt{1 + \mu^2(\boldsymbol{A})}}{2\mu(\boldsymbol{A})\sqrt{m}},$$

which can be rewritten as in the statement of the Theorem.

It remains to prove that $\boldsymbol{\phi}_0$ is also the unique solution to (10) in the case where this condition is satisfied and $\boldsymbol{\phi}_0$ is the unique solution to (9). To see this, note that $\boldsymbol{\phi}_0 = \phi(\boldsymbol{x}_0)$ is by definition a feasible point of (10). In addition, the feasible set of (10) is included in the one of (9), while the problems (9) and (10) share the same cost function. Thus, if $\boldsymbol{\phi}_0$ is the unique solution to (9), there cannot be another feasible $\boldsymbol{\phi} \neq \boldsymbol{\phi}_0$ with lower or equal cost function value for (10).

$\square$

Corollary 1. *Let $\boldsymbol{x}_0$ be a solution to the polynomial equations (2) and m be the number of components of ϕ associated to a variable. If the condition*

$$\|\boldsymbol{x}_0\|_0 < \frac{1}{2\sqrt{m}}\sqrt{1 + \frac{1}{\mu^2(\boldsymbol{A})}}$$

holds, then $\boldsymbol{x}_0$ is the unique solution to the minimization problem (1)–(2) and it can be computed as $\boldsymbol{x}_0 = \phi^{-1}(\hat{\boldsymbol{\phi}})$ with $\hat{\boldsymbol{\phi}}$ the solution to either (9) or (10).

Proof. Assume there exists another solution $\boldsymbol{x}_1 \neq \boldsymbol{x}_0$ to (1)–(2), and thus with $\|\boldsymbol{x}_1\|_0 \leq \|\boldsymbol{x}_0\|_0$. Then, Theorem 3 implies that both $\phi(\boldsymbol{x}_1)$ and $\phi(\boldsymbol{x}_0)$ are *unique* solutions to either (9) or (10), and thus that $\phi(\boldsymbol{x}_1) = \phi(\boldsymbol{x}_0) = \hat{\boldsymbol{\phi}}$ (where $\hat{\boldsymbol{\phi}}$ is similarly defined as the solution to either (9) or (10) in this case). But this contradicts the definition of the mapping ϕ implying $\phi(\boldsymbol{x}_1) \neq \phi(\boldsymbol{x}_0)$ whenever $\boldsymbol{x}_1 \neq \boldsymbol{x}_0$. Therefore, the assumption $\boldsymbol{x}_1 \neq \boldsymbol{x}_0$ cannot hold and $\boldsymbol{x}_0$ is the unique solution to (1)–(2), while $\phi^{-1}(\hat{\boldsymbol{\phi}}) = \phi^{-1}(\phi(\boldsymbol{x}_0)) = \boldsymbol{x}_0$. $\qquad\square$

Theorem 4. *Let $\hat{\boldsymbol{\phi}}$ be a solution to (9) and m be the number of components of ϕ associated to a variable. If the condition*

$$\|\{\boldsymbol{W}_j\hat{\boldsymbol{\phi}}\}_{j=1}^n\|_0 < \frac{1}{2\sqrt{m}}\sqrt{1 + \frac{1}{\mu^2(\boldsymbol{A})}}$$

holds, then $\hat{\boldsymbol{\phi}}$ is the unique solution to both (8) and (9). If, in addition, $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}})$ satisfies the polynomial constraints (2), then $\hat{\boldsymbol{x}}$ is the unique solution to (1)–(2).

Proof. By following steps similar to those in the proof of Theorem 3 with $\hat{\boldsymbol{\phi}}$ instead of $\boldsymbol{\phi}_0$, we obtain (by replacing $\|\boldsymbol{x}_0\|_0$ by $\|\{\boldsymbol{W}_j\hat{\boldsymbol{\phi}}\}_{j=1}^n\|_0$ in (14)) that $\hat{\boldsymbol{\phi}}$ is the unique solution to (9). Then, let $\boldsymbol{\phi}^*$ be a solution to (8). This implies $\|\{\boldsymbol{W}_j\boldsymbol{\phi}^*\}_{j=1}^n\|_0 \leq \|\{\boldsymbol{W}_j\hat{\boldsymbol{\phi}}\}_{j=1}^n\|_0$ and, under the condition of the Theorem,

$$\|\{\boldsymbol{W}_j\boldsymbol{\phi}^*\}_{j=1}^n\|_0 < \frac{1}{2\sqrt{m}}\sqrt{1 + \frac{1}{\mu^2(\boldsymbol{A})}}. \tag{15}$$

By following similar steps again, we obtain that $\boldsymbol{\phi}^*$ is also the unique solution to (9) and thus that $\boldsymbol{\phi}^* = \hat{\boldsymbol{\phi}}$ is the unique solution to (8).

Finally, since $\hat{\boldsymbol{\phi}}$ is the unique solution to (8), if $\hat{\boldsymbol{x}}$ satisfies the polynomial equations (2), Theorem 2 implies that $\hat{\boldsymbol{x}}$ is the solution to (1)–(2). $\qquad\square$

In comparison with Theorem 3, Theorem 4 provides a condition that only depends on the estimate obtained by solving (9) rather than on the sought solution.

2.4 Enhancing sparsity

In practice, convex relaxations such as the ones described above provide a good step towards the solution but might fail to yield the exact solution with sufficient sparsity. In such cases, it is common practice to improve the sparsity of the solution by repeating the procedure with a well-chosen weighting of the variables as described in [Candès et al., 2008, Le et al., 2013]. These techniques can be directly applied to improve the ℓ_1-minimization method of Sect. 2.1 while they are adapted below to group-sparsity as considered in Sect. 2.2.

2.4.1 Iterative reweighting

The classical reweighting scheme of [Candès et al., 2008] for sparse recovery improves the sparsity of the solution by solving a sequence of linear programs. It can be adapted to the group-sparse

recovery problem by iteratively solving

$$\hat{\boldsymbol{\phi}} = \arg \min_{\boldsymbol{\phi} \in \mathbb{R}^M} \sum_{j=1}^n \mu_j \|\boldsymbol{W}_j \boldsymbol{\phi}\|_2 \tag{16}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{y} - \boldsymbol{b}$$

$$\phi_k \geq 0, \ \forall k \text{ such that } (\alpha_k)_j \text{ is even for all } j,$$

with weights μ_j initially set to 1 and refined at each iteration by

$$\mu_j = \frac{1}{\|\boldsymbol{W}_j \hat{\boldsymbol{\phi}}\|_2 + \epsilon}$$

for a given small value of $\epsilon > 0$.

The basic idea is to decrease the influence of groups of variables with large ℓ_2-norms that are assumed to be nonzero in the solution while increasing the weight of groups with small norms in order to force them towards zero.

2.4.2 Selective ℓ_1/ℓ_2-minimization

The $\text{S}\ell_1\text{M}$ algorithm proposed in [Le et al., 2013] is another reweighted ℓ_1-minimization mechanism which sets a single weight at zero at each iteration. Though typically requiring more computation time than the previous approach due to a number of iterations equal to the number of non-zero elements, this algorithm can recover sparse solutions in cases where the classical reweighting scheme of [Candès et al., 2008] fails. Other advantages include the absence of a tuning parameter and the presence of a convergence analysis [Le et al., 2013]. The $\text{S}\ell_1\ell_2\text{M}$ algorithm below is an adaptation of $\text{S}\ell_1\text{M}$ to group-sparse problems.

1. Initialize all weights $\mu_j = 1$, $j = 1, \ldots, n$.

2. Solve the weighted group-sparse problem (16).

3. Find $k \in \arg\max_{j \in \{1, \ldots, n\}} \|\boldsymbol{W}_j \hat{\boldsymbol{\phi}}\|_2$.

4. Set $\mu_k = 0$ (to relax the sparsity constraint on the kth group).

5. Repeat from Step 2 until $\sum_{j=1}^n \mu_j \|\boldsymbol{W}_j \hat{\boldsymbol{\phi}}\|_2 = 0$.

Note that in this algorithm, the number of iterations is equal to the number of nonzero groups[3], which, for polynomial basis pursuit, is $\|\boldsymbol{x}_0\|_0$ and is typically small. This results in a fast and accurate method for polynomial basis pursuit, as will be seen in Sect. 6.

3 Greedy approach

As mentioned in the introduction, there are two major techniques to minimize an ℓ_0-pseudo-norm: the BP approach and the greedy approach. We now consider the second one to solve problem (8), and more particularly develop two variants of the greedy approach: the exact method and the approximate method. The exact method is intended for small-scale problems where the number of possible combinations of base variables remains small. The approximate method is designed to circumvent this limitation and applies to much larger problems.

Exact greedy algorithm. The exact method is implemented as follows, where we let $\epsilon = 0$ if the polynomial system of equations is assumed to be feasible, and $\epsilon > 0$ otherwise.

1. Initialize: $\hat{n} = 0$ and $e = +\infty$.

2. $\hat{n} \leftarrow \hat{n} + 1$.

---

[3]The maximal number of iterations is the number of groups n, but if the correct sparsity pattern is recovered then the algorithm stops earlier.

3. For all combinations C of $\hat{n}$ variables among n:

   (a) Set $S = \{1, \ldots, M\} \setminus \bigcup_{k \notin C} I_k$, where the index sets I_k are defined as in (6).

   (b) Build the submatrix $\boldsymbol{A}_S$ with the columns of $\boldsymbol{A}$ with index in S.

   (c) Solve
   $$\boldsymbol{\phi}_S = \arg \min_{\boldsymbol{\phi} \in \mathbb{R}^{|S|}} \|\boldsymbol{A}_S \boldsymbol{\phi} + \boldsymbol{b} - \boldsymbol{y}\|_2^2.$$

   (d) Update $e \leftarrow \min\{e, \ \|\boldsymbol{A}_S \boldsymbol{\phi}_S + \boldsymbol{b} - \boldsymbol{y}\|_2^2\}$.

   (e) If $e \leq \epsilon$, compute $\hat{\boldsymbol{\phi}}$ by setting its components of index in S to the values in $\boldsymbol{\phi}_S$ and the others to 0. Return $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}})$.

4. If $\hat{n} < n$, repeat from Step 2, otherwise return an infeasibility certificate.

In Step 3.(a), S corresponds to the support of $\phi(\boldsymbol{x})$ when $supp(\boldsymbol{x}) = C$, where C is a combination of $\hat{n}$ indexes from 1 to n. The maximal number of least squares (LS) problems to solve in Step 3.(c) is 2^n. But many of these are spared by starting with the sparsest combinations and stopping as soon as a solution is found. Thus, if a sparse solution $\boldsymbol{x}_0$ with $\|\boldsymbol{x}_0\|_0 \leq n/2$ exists, only $\sum_{t=1}^{\|\boldsymbol{x}_0\|_0} \binom{n}{t} \leq \|\boldsymbol{x}_0\|_0 \binom{n}{\|\boldsymbol{x}_0\|_0}$ LS problems are solved. At iteration t, the LS problem is of size $N \times |S|$, with $|S| = \sum_{q=1}^{d} \binom{t+q-1}{q} \leq d \binom{t+d-1}{d}$. Thus, assuming a complexity $\mathcal{LS}(N_1, N_2)$ for an LS problem of size $N_1 \times N_2$, the overall complexity of the algorithm scales as $\|\boldsymbol{x}_0\|_0 \binom{n}{\|\boldsymbol{x}_0\|_0} \mathcal{LS}\left(N, d\left(\frac{\|\boldsymbol{x}_0\|_0 + d - 1}{d}\right)\right)$, which is exponential in n, d and $\|\boldsymbol{x}_0\|_0$.

With $\epsilon = 0$, the exact greedy algorithm above can be slightly modified to compute all the solutions to (8), simply by letting the *for* loop in Step 3 complete instead of returning as soon as a solution is found. As a result, the algorithm could provide a uniqueness certificate for the solution of (8) from which Theorem 2 could be applied to conclude that the solution coincides with the unique minimizer of (1)–(2).

Approximate greedy algorithm. The approximate method is similar except that it explores only a single branch of the tree of possible combinations. Its implementation uses a set S of retained variables (more precisely, S contains the indexes of these variables):

1. Initialize the set of nonzero variables: $S = \emptyset$.

2. For all $j \in \{1, \ldots, n\} \setminus S$,

   (a) Set $S_j = \{1, \ldots, M\} \setminus \bigcup_{k \notin S \cup \{j\}} I_k$, where the index sets I_k are defined as in (6).

   (b) Build the submatrix $\boldsymbol{A}_{S_j}$ with the columns of $\boldsymbol{A}$ with index in S_j.

   (c) Solve
   $$\boldsymbol{\phi}_j = \arg \min_{\boldsymbol{\phi} \in \mathbb{R}^{|S_j|}} \|\boldsymbol{A}_{S_j} \boldsymbol{\phi} + \boldsymbol{b} - \boldsymbol{y}\|_2^2.$$

3. Select the variable that minimizes the error if added to S:
   $$k = \arg \min_{j \in \{1, \ldots, n\}} \|\boldsymbol{A}_{S_j} \boldsymbol{\phi}_j + \boldsymbol{b} - \boldsymbol{y}\|_2^2.$$

4. Update $S \leftarrow S \cup \{k\}$.

5. Repeat from Step 2 until $\|\boldsymbol{A}_{S_k} \boldsymbol{\phi}_k + \boldsymbol{b} - \boldsymbol{y}\|_2^2 \leq \epsilon$.

6. Compute $\hat{\boldsymbol{\phi}}$ by setting its components of index in S_k to the values in $\boldsymbol{\phi}_k$ and the others to 0.

7. Return $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}})$ and the error $\|\boldsymbol{A}_{S_k} \boldsymbol{\phi}_k + \boldsymbol{b} - \boldsymbol{y}\|_2^2$.

The algorithm starts with an empty set of nonzero variables S and adds a single variable to that set at each iteration. The variable retained at a given iteration is the one that, if added, leads to the minimum sum of squared errors for the equations $\boldsymbol{A}_{S_j}\boldsymbol{\phi}_j = \boldsymbol{y} - \boldsymbol{b}$. In Step 2.(a), S_j corresponds to the support of $\phi(\boldsymbol{x})$ when $supp(\boldsymbol{x}) = S \cup \{j\}$. Note that the value of the minimizer $\boldsymbol{\phi}_k$ is not retained but re-estimated at the next iteration. The reason for this is that there is no guarantee that the components of $\boldsymbol{\phi}_k$ correspond to monomials of base variables.

Since one base variable is added at each iteration t, the number of iterations equals the sparsity of the returned solution and $t \le \|\hat{\boldsymbol{x}}\|_0$. In this approximate variant of the algorithm, each iteration requires solving only $(n - t + 1)$ LS problems, which yields a total number of LS problems equal to $\sum_{t=1}^{\|\hat{\boldsymbol{x}}\|_0}(n - t + 1) \le n\|\hat{\boldsymbol{x}}\|_0$. Then, bounding the complexity of each LS problem as for the exact greedy algorithm leads to an overall complexity bounded by $n\|\hat{\boldsymbol{x}}\|_0 \mathcal{LS}\left(N, d\binom{\|\hat{\boldsymbol{x}}\|_0 + d - 1}{d}\right)$.

Since we do not have a result on the convergence of the algorithm to the sparsest solution, we cannot bound $\|\hat{\boldsymbol{x}}\|_0$ except by n and the worst-case complexity is exponential in n. However, in practice, if the algorithm finds a sparse solution $\hat{\boldsymbol{x}}$, the complexity is exponential in its sparsity $\|\hat{\boldsymbol{x}}\|_0$, but only linear in n. This is rather promising given the empirical results to be shown in Sect. 6 which suggest that this occurs with high probability.

4   Purely nonlinear polynomials

We now consider the case of purely nonlinear polynomials $p_i^d(\boldsymbol{x})$ without a linear part for some variables, i.e., with $a_{ik} = 0$, $i = 1, \ldots, N$, for some $k \in \{1, \ldots, n\}$ (according to the ordering of the multi-indexes, the linear monomials correspond to the first coefficients with $1 \le k \le n$). For this specific case where Assumption 1 does not hold, some of the first n columns of $\boldsymbol{A}$ are zero and the corresponding components of $\boldsymbol{\phi}$ are unconstrained, thus set to arbitrary values in $\hat{\boldsymbol{\phi}}$. As a result, not only does the analysis of Sect. 2.3 not hold, but the estimate $\hat{\boldsymbol{x}}$ cannot be directly obtained by the inverse mapping ϕ^{-1} as the first components of $\hat{\boldsymbol{\phi}}$.

However, the core of the method remains applicable to purely nonlinear polynomials. More precisely, we can solve (5), (9) or (10) (or apply a reweighting scheme of Sect. 2.4) to obtain $\hat{\boldsymbol{\phi}}$ and the corresponding support of $\hat{\boldsymbol{x}}$ as

$$supp(\hat{\boldsymbol{x}}) = \{j \in \{1, \ldots, n\} : \|\boldsymbol{W}_j \hat{\boldsymbol{\phi}}\|_2 \ne 0\},$$

while the greedy algorithms of Sect. 3 directly estimate $supp(\hat{\boldsymbol{x}})$. Then, the estimate $\hat{\boldsymbol{x}}$ can be computed from the higher-degree monomials as, e.g., $\hat{x}_j = \pm\sqrt[q]{\hat{\phi}_{jq}}$, where the subscript $_{jq}$ denotes the index such that $(\phi(\cdot))_{jq} : \boldsymbol{x} \mapsto x_j^q$. The precise procedure to compute $\hat{\boldsymbol{x}}$ actually depends on the monomials involved in the polynomials. The most straightforward manner is to compute $\hat{x}_j$ from the estimate of its smallest nonzero odd power:

$$\forall j \in supp(\hat{\boldsymbol{x}}), \quad \hat{x}_j = \sqrt[2\hat{p}+1]{\hat{\phi}_{j(2\hat{p}+1)}}, \quad \text{with } \hat{p} = \min_{p \in \{0, \ldots, (d-1)/2\}} p, \text{ s.t. } \hat{\phi}_{j(2p+1)} \ne 0. \quad (17)$$

But for polynomial systems that involve only monomials with even degrees, the minimization computing $\hat{p}$ in (17) has no solution and the procedure is slightly more complex. For instance, with purely quadratic equations, the absolute value of $\hat{x}_j$ is given by $|\hat{x}_j| = \sqrt{\hat{\phi}_{j2}}$ and the sign must be determined by looking at the signs of the estimates of the bilinear terms $\widehat{x_i x_j}$, $i \ne j$. In addition, note that for such cases, the solution of (1)–(2) is not unique for symmetry reasons and the method cannot be analyzed as in Sect. 2.3 in terms of convergence towards the sparsest solution. Then, a different notion of uniqueness is usually considered in the literature dedicated to purely quadratic equations [Balan et al., 2006, Bandeira et al., 2014, Ohlsson and Eldar, 2013, Ranieri et al., 2013].

5   Polynomial denoising

In many applications, the equations $y_i = f_i(\boldsymbol{x})$ need to be relaxed to an error-tolerant form for various reasons, which can for instance be interpreted as having access to noisy measurements,

$y_i = f_i(\boldsymbol{x}) + e_i$, with unknown noise terms e_i. In this case, we reformulate the general problem (1) as a denoising one:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{e}\in\mathbb{R}^N} \|\boldsymbol{x}\|_0 \tag{18}$$

$$\text{s.t. } y_i = f_i(\boldsymbol{x}) + e_i, \quad i = 1, \ldots, N,$$

$$\|\boldsymbol{e}\|_p \le \varepsilon,$$

where $\varepsilon > 0$ is a fixed threshold on the noise ℓ_p-norm, with typical choices for p being $p = 1$, $p = 2$ or $p = \infty$.

With polynomial constraints,

$$y_i = f_i(\boldsymbol{x}) + e_i = p_i^d(\boldsymbol{x}) + e_i = b_i + \sum_{k=1}^{M} a_{ik}\boldsymbol{x}^{\alpha_k} + e_i, \quad i = 1, \ldots, N, \tag{19}$$

convex relaxations similar to the ones described in Sect. 2 can be derived to obtain polynomial BP denoising methods. This leads to solving

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^M} \|\boldsymbol{W}\boldsymbol{\phi}\|_1 \tag{20}$$

$$\text{s.t. } \|\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{b} - \boldsymbol{y}\|_p \le \varepsilon,$$

for the ℓ_1-minimization method and

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^M} \sum_{j=1}^{n} \|\boldsymbol{W}_j\boldsymbol{\phi}\|_2 \tag{21}$$

$$\text{s.t. } \|\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{b} - \boldsymbol{y}\|_p \le \varepsilon$$

with ℓ_1/ℓ_2-minimization. For $p \in \{1, \infty\}$, problem (20) remains a linear program, while $p = 2$ leads to a SOCP. Problem (21) can still be written as a SOCP for all $p \in \{1, 2, \infty\}$ and be solved by the generic solvers that apply to (9). Thus, the enhancements proposed in Sect. 2 for the formulations (5) and (9) (such as reweighting schemes or the addition of structural constraints) can be easily transposed to the noisy case.

Regarding the greedy algorithms of Sect. 3, they are already applicable to the noisy case, for which it suffices to set an appropriate threshold ε on the noise ℓ_2-norm. Adaptations of these algorithms to $p = 1$ or $p = \infty$ are straightforward, but require solving a convex optimization problem in sub-step (c) without a closed-form solution and thus without the same computational benefit for the approximate greedy approach.

5.1 Stability analysis

In the noisy case, the solution to (18) is in general not unique and the analysis focuses on stability rather than on exact recovery. The following theorem provides such a stability result for the ℓ_1-minimization.

Theorem 5. *Let $(\boldsymbol{x}_0, \boldsymbol{e}_0)$ denote a solution to (18)–(19) for $p = 2$. If the inequality*

$$\|\boldsymbol{x}_0\|_0 < \frac{n}{4M}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right) \tag{22}$$

holds, then $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\boldsymbol{\phi}})$ with $\hat{\boldsymbol{\phi}}$ the solution to (20) for $p = 2$ must obey

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2^2 \le \frac{4\varepsilon^2}{1 - \mu(\boldsymbol{A})(4M\|\boldsymbol{x}_0\|_0/n - 1)}. \tag{23}$$

Proof. Assume (18)–(19) has a solution $(\boldsymbol{x}_0, \boldsymbol{e}_0)$. Then, $\boldsymbol{A}\boldsymbol{\phi}_0 = \boldsymbol{y} - \boldsymbol{b} - \boldsymbol{e}_0$ with $\boldsymbol{\phi}_0 = \phi(\boldsymbol{x}_0)$ and $\|\boldsymbol{e}_0\|_2 \le \varepsilon$. By Proposition 1, we have $\|\boldsymbol{\phi}_0\|_0 \le \frac{M}{n}\|\boldsymbol{x}_0\|_0$. But, according to Theorem 9 in [Bruckstein et al., 2009], if $\|\boldsymbol{\phi}_0\|_0 < \frac{1}{4}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$ then $\hat{\boldsymbol{\phi}}$ must obey

$$\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}_0\|_2^2 \le \frac{4\varepsilon^2}{1 - \mu(\boldsymbol{A})(4\|\boldsymbol{\phi}_0\|_0 - 1)}. \tag{24}$$

Thus, if (22) holds, so does (24) and $\|\hat{\phi} - \phi_0\|_2^2 \le 4\varepsilon^2/[1 - \mu(\boldsymbol{A})(4M\|\boldsymbol{x}_0\|_0/n-1)]$. Since $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2^2 \le \|\hat{\phi} - \phi_0\|_2^2$, this completes the proof.

□

For the ℓ_1/ℓ_2-minimization, we have the following stability result.

Theorem 6. *Let $(\boldsymbol{x}_0, \boldsymbol{e}_0)$ denote a solution to (18)–(19) for $p = 2$. If the inequality*

$$\|\boldsymbol{x}_0\|_0 < \frac{1}{4nM}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right) \tag{25}$$

holds, then $\hat{\boldsymbol{x}} = \phi^{-1}(\hat{\phi})$ with $\hat{\phi}$ the solution to (21) for $p = 2$ must obey

$$\|\boldsymbol{W}_x(\hat{\boldsymbol{x}} - \boldsymbol{x}_0)\|_2^2 \le \frac{4n\varepsilon^2}{1 - \mu(\boldsymbol{A})(4nM\|\boldsymbol{x}_0\|_0 - 1)}, \tag{26}$$

where $\boldsymbol{W}_x = diag(\|\boldsymbol{A}_1\|_2, \ldots, \|\boldsymbol{A}_n\|_2)$.

Proof. Assume (18)–(19) has a solution $(\boldsymbol{x}_0, \boldsymbol{e}_0)$. Let us define $\phi_0 = \phi(\boldsymbol{x}_0)$ and $\boldsymbol{\delta} = \hat{\phi} - \phi_0$. To prove the theorem, we will first bound from above the norm $\|\boldsymbol{W}\boldsymbol{\delta}\|_2$, where $\boldsymbol{W}$ is the diagonal matrix of precompensating weights. This part of the proof follows a path similar to that of Theorem 3.1 in [Donoho et al., 2006], while adapting it to the group-sparse setting and mixed ℓ_p/ℓ_q norms.

Due to the definition of $\hat{\phi}$ as a minimizer of (21), $\boldsymbol{\delta}$ must satisfy either

$$\sum_{j=1}^n \|\boldsymbol{W}_j(\phi_0 + \boldsymbol{\delta})\|_2 < \sum_{j=1}^n \|\boldsymbol{W}_j\phi_0\|_2$$

or $\boldsymbol{\delta} = \boldsymbol{0}$, in which case the statement is obvious. The inequality above can be rewritten as

$$\sum_{j \in I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 + \sum_{j \notin I_0} \|\boldsymbol{W}_j(\phi_0 + \boldsymbol{\delta})\|_2 - \|\boldsymbol{W}_j\phi_0\|_2 < 0,$$

where $I_0 = \{j \in \{1, \ldots, n\} : \boldsymbol{W}_j\phi_0 = \boldsymbol{0}\}$. By the triangle inequality, $\|\boldsymbol{u} + \boldsymbol{v}\|_2 - \|\boldsymbol{u}\|_2 \ge -\|\boldsymbol{v}\|_2$, this implies

$$\sum_{j \in I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 - \sum_{j \notin I_0} \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2 < 0. \tag{27}$$

In addition, $\boldsymbol{\delta}$ must satisfy the constraints in (21) as

$$\|\boldsymbol{A}(\phi_0 + \boldsymbol{\delta}) + \boldsymbol{b} - \boldsymbol{y}\|_2 \le \varepsilon,$$

in which $\boldsymbol{y}$ can be replaced by $\boldsymbol{A}\phi_0 + \boldsymbol{b} + \boldsymbol{e}_0$, leading to

$$\|\boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{e}_0\|_2 \le \varepsilon.$$

Using $\|\boldsymbol{u}\|_2 \le \|\boldsymbol{u} - \boldsymbol{v}\|_2 + \|\boldsymbol{v}\|_2$, this implies $\|\boldsymbol{A}\boldsymbol{\delta}\|_2 \le 2\varepsilon$, which further gives

$$4\varepsilon^2 \ge \|\boldsymbol{A}\boldsymbol{\delta}\|_2^2 = \|\boldsymbol{A}\boldsymbol{W}^{-1}\boldsymbol{W}\boldsymbol{\delta}\|_2 = (\boldsymbol{W}\boldsymbol{\delta})^T\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1}(\boldsymbol{W}\boldsymbol{\delta})$$

$$= \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 + (\boldsymbol{W}\boldsymbol{\delta})^T(\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I})(\boldsymbol{W}\boldsymbol{\delta})$$

$$\ge \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 - \left|(\boldsymbol{W}\boldsymbol{\delta})^T(\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I})(\boldsymbol{W}\boldsymbol{\delta})\right|$$

$$\ge \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 - |\boldsymbol{W}\boldsymbol{\delta}|^T|\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I}||\boldsymbol{W}\boldsymbol{\delta}|$$

$$\ge \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 - \mu(\boldsymbol{A})(\|\boldsymbol{W}\boldsymbol{\delta}\|_1^2 - \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2)$$

$$= (1 + \mu(\boldsymbol{A}))\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 - \mu(\boldsymbol{A})\|\boldsymbol{W}\boldsymbol{\delta}\|_1^2 \tag{28}$$

where we used $\boldsymbol{W}^{-1}\boldsymbol{W} = \boldsymbol{I}$ and the fact that the diagonal entries of $|\boldsymbol{W}^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{-1} - \boldsymbol{I}|$ are zeros while off-diagonal entries are bounded from above by $\mu(\boldsymbol{A})$.

Due to $\boldsymbol{W}_j\boldsymbol{\delta}$ being a vector with a subset of entries from $\boldsymbol{W}\boldsymbol{\delta}$, we have $\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \geq \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2^2$, $j = 1,\ldots,n$, and thus

$$\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \geq \frac{1}{n}\sum_{j=1}^n \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2^2 = \frac{1}{n}\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2. \tag{29}$$

Since the groups defined by the $\boldsymbol{W}_j$'s overlap, $\|\boldsymbol{W}\boldsymbol{\delta}\|_2 \leq \sum_{j=1}^n \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2$, and the squared ℓ_1-norm in (28) can be bounded by

$$\|\boldsymbol{W}\boldsymbol{\delta}\|_1^2 \leq M\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \leq M\left(\sum_{j=1}^n \|\boldsymbol{W}_j\boldsymbol{\delta}\|_2\right)^2 = M\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{1,2}^2. \tag{30}$$

Introducing the bounds (29)–(30) in (28) yields

$$\frac{1+\mu(\boldsymbol{A})}{n}\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2 - \mu(\boldsymbol{A})M\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{1,2}^2 \leq 4\varepsilon^2. \tag{31}$$

We will now use this inequality to derive an upper bound on $\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2$, which will also apply to $\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \leq \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2$, since the groups overlap and the squared components of $\boldsymbol{W}\boldsymbol{\delta}$ are summed multiple times in $\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2$. To derive the upper bound, we first introduce a few notations:

$$a = \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\in I_0}\|_{1,2}, \quad b = \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\notin I_0}\|_{1,2},$$

and

$$c_0 = \left(\frac{\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\in I_0}\|_{2,2}}{\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\in I_0}\|_{1,2}}\right)^2 \in \left[\frac{1}{|I_0|},1\right], \quad c_1 = \left(\frac{\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\notin I_0}\|_{2,2}}{\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\notin I_0}\|_{1,2}}\right)^2 \in \left[\frac{1}{n-|I_0|},1\right],$$

where the box bounds are obtained by classical relations between the ℓ_1 and ℓ_2 norms ($\forall \boldsymbol{u} \in \mathbb{R}^k$, $\|\boldsymbol{u}\|_2 \leq \|\boldsymbol{u}\|_1 \leq \sqrt{k}\|\boldsymbol{u}\|_2$). With these notations, the term to bound is rewritten as

$$\|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2 = \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\in I_0}\|_{2,2}^2 + \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j\notin I_0}\|_{2,2}^2 = c_0 a^2 + c_1 b^2,$$

while the inequality (31) becomes

$$\frac{1+\mu(\boldsymbol{A})}{n}(c_0 a^2 + c_1 b^2) - \mu(\boldsymbol{A})M(a+b)^2 \leq 4\varepsilon^2.$$

We further reformulate this constraint by letting $a = \rho b$:

$$\frac{1+\mu(\boldsymbol{A})}{n}(c_0\rho^2 + c_1)b^2 - \mu(\boldsymbol{A})M(1+\rho)^2 b^2 \leq 4\varepsilon^2. \tag{32}$$

Let $\gamma = (1+\rho)^2/(c_0\rho^2 + c_1)$. Due to (27), we have $a < b$ and thus $\rho \in [0,1)$, which, together with the bounds on c_0 and c_1, gives the constraints $1 \leq \gamma \leq 4(n-|I_0|)$. By setting $V = (c_0\rho^2 + c_1)b^2$, (31) is rewritten as

$$\frac{1+\mu(\boldsymbol{A})}{n}V - \mu(\boldsymbol{A})M\gamma V \leq 4\varepsilon^2,$$

where

$$\frac{1+\mu(\boldsymbol{A})}{n} - \mu(\boldsymbol{A})M\gamma \geq \frac{1+\mu(\boldsymbol{A})}{n} - 4(n-|I_0|)\mu(\boldsymbol{A})M > 0,$$

since $\gamma \leq 4(n-|I_0|)$ and the positivity is ensured by the condition (25) and the fact that $\|\boldsymbol{x}_0\|_0 = n - |I_0|$. Thus,

$$\|\boldsymbol{W}\boldsymbol{\delta}\|_2^2 \leq \|\{\boldsymbol{W}_j\boldsymbol{\delta}\}_{j=1}^n\|_{2,2}^2 = V \leq \frac{4n\varepsilon^2}{1+\mu(\boldsymbol{A}) - 4\mu(\boldsymbol{A})nM\|\boldsymbol{x}_0\|_0}$$

and, since $\|\boldsymbol{W}_x(\hat{\boldsymbol{x}} - \boldsymbol{x}_0)\|_2^2 \leq \|\boldsymbol{W}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}_0)\|_2^2 = \|\boldsymbol{W}\boldsymbol{\delta}\|_2^2$, (26) is proved.

$\square$

Table 1: Results on the example from [Ohlsson et al., 2013b] with quadratic equations.

Method	IHT	QBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Success rate	54%	79%	0%	0%	97%	97%	91%	100%
Mean time (s)	1.92	1.76	0.26	0.07	0.46	0.15	0.03	0.14

Table 2: Results on the example from [Ohlsson et al., 2013a] with polynomials of degree $d = 4$.

Method	IHT	NLBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Success rate	66%	100%	85%	16%	100%	100%	100%	100%
Mean time (s)	0.43	13.1	0.22	0.06	0.42	0.08	0.006	0.006

6 Experiments

This section evaluates the efficiency of the BP and greedy methods in terms of accuracy and computing time for the noiseless case in Sect. 6.1 and the noisy case in Sect. 6.2. Here, the definition of accuracy depends on the presence of noise in the equations, while the computing time refers to Matlab implementations[4] using MOSEK and CVX for the convex programs and running on a standard laptop (except for times reported in Fig. 3).

The following methods are compared: the iterative hard thresholding (IHT) algorithm [Blumensath, 2013] as implemented by [Beck and Eldar, 2013], the quadratic (QBP) and nonlinear (NLBP) BP methods of [Ohlsson et al., 2013b, Ohlsson et al., 2013a], the simple ℓ_1-minimization (ℓ_1M) solving (5)[5], the ℓ_1/ℓ_2-minimization ($\ell_1\ell_2$M) solving (10) with its iteratively reweighted counterpart (IR$\ell_1\ell_2$M) using 10 iterations as defined in Sect. 2.4.1 for $\epsilon = 0.001$, the selective ℓ_1/ℓ_2-minimization (S$\ell_1\ell_2$M) of Sect. 2.4.2, and the exact (EGA) and approximate (AGA) greedy algorithms of Sect. 3. For the noisy cases, error tolerant formulations of these methods as described in Sect. 5 are used.

6.1 Exact recovery with noiseless equations

In the noiseless setting considered in the following experiments, accuracy is defined as the ability of a method to recover the sparsest solution $\boldsymbol{x}_0$ of a polynomial system and is measured as a success rate, i.e., the percentage of systems for which it recovers $\boldsymbol{x}_0$ (meaning $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2 \leq 10^{-6}$) in a Monte Carlo experiment involving 100 trials with random polynomial coefficients but same $\boldsymbol{x}_0$. More precisely, for each experiment, given a sparsity level $\|\boldsymbol{x}_0\|_0$, we generate an $\boldsymbol{x}_0$ with the first $\|\boldsymbol{x}_0\|_0$ components set to 1 and the rest at 0, while the values of y_i, $i = 1, \ldots, N$, are given by (2) with b_i and a_{ik}, $k = 1, \ldots, M$, randomly drawn for each trial according to a zero-mean Gaussian distribution with unit variance.

Sparse solutions of quadratic equations. Consider example A in [Ohlsson et al., 2013b] with $N = 25$ quadratic equations ($d = 2$) in $n = 20$ variables. The true $\boldsymbol{x}_0$ that we aim at recovering has three components at 1 and the rest at 0: $\|\boldsymbol{x}_0\|_0 = 3$. The results reported in Table 1 show that all reweighted BP and greedy methods recover the solution in almost all trials in less than 1 second, whereas the IHT and QBP methods lead to more failures despite longer computing times. However, the ℓ_1M method, which does not enforce the polynomial structure on $\hat{\boldsymbol{\phi}}$, cannot recover the correct solution. The straightforward optimization of the ℓ_1/ℓ_2-norm ($\ell_1\ell_2$M) also fails, which shows the importance of reweighting for obtaining truly sparse solutions.

Higher-degree polynomial equations. Consider now example A in [Ohlsson et al., 2013a] with $N = 50$ polynomial equations of degree $d = 4$ in $n = 5$ variables with two nonzeros ($\|\boldsymbol{x}_0\|_0 =$

Table 3: Results with purely quadratic equations.

Method	QBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Success rate	0%	0%	3%	100%	99%	91%	100%
Mean time (s)	1.42	0.27	0.03	0.77	0.19	0.05	0.36

Table 4: Results with purely nonlinear equations of degree $d = 4$.

Method	NLBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Success rate	100%	0%	15%	100%	100%	100%	100%
Mean time (s)	13.1	0.21	0.07	0.70	0.12	0.01	0.012

Table 5: Results with purely quadratic equations generated as in phase retrieval.

Method	GSS	QBP	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Success rate	32%	0%	79%	72%	71%	100%
Mean time (s)	0.10	1.45	0.71	0.25	0.05	0.38

2). Though using higher degree polynomials, according to the results shown in Table 2, this problem setting seems easier than the quadratic example above. Indeed, the simple ℓ_1M method already obtains a success rate of 85% while all reweighted BP and greedy methods achieve a 100% success rate. This is due to the increase of the number of equations, N, and the decrease of the number of variables, n, as will be emphasized by additional experiments below. Here, S$\ell_1\ell_2$M can be much faster than IR$\ell_1\ell_2$M based on the iterative reweighting of [Candès et al., 2008] (Sect. 2.4.1) because the number of nonzero groups is very small: $2 < n <$ typical number of iterations for the reweighting of [Candès et al., 2008]. However, the AGA obtains the same success rate, while being one order of magnitude faster.

The NLBP method also recovered the correct solution in all trials, but with a much longer computing time. This time is roughly divided in halves for the construction of quadratic constraints that can be handled by QBP on the one hand and the semi-definite optimization of QBP on the other hand. Here again, the IHT method offers a low success rate, though it uses additional information on $\|\boldsymbol{x}_0\|_0$ to compute $\hat{\boldsymbol{x}}$ with the optimal sparsity level (which is unknown to the other methods). This is due in part to the difficulty of tuning the gradient step-size to obtain convergence with an objective function whose gradient is not Lipschitz continuous. For this reason, we exclude the IHT method from the remaining experiments.

Purely nonlinear equations. Consider now the particular case of purely nonlinear equations discussed in Sect. 4. Given $N = 25$ purely quadratic polynomials $p_i^2(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{Q}_i \boldsymbol{x}$ in $n = 20$ variables, we compute the values $y_i = p_i^2(\boldsymbol{x}_0)$. Results are shown in Table 3 for $\|\boldsymbol{x}_0\|_0 = 3$ and where the definition of the success rate is slightly modified: since the sparsest solution is not unique in this case, a successful trial is defined as one for which the estimate $\hat{\boldsymbol{x}}$ belongs to the set of solutions (meaning that $\min\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2, \|\hat{\boldsymbol{x}} + \boldsymbol{x}_0\|_2\} \leq 10^{-6}$). Table 4 reports results of similar experiments with a higher degree $d = 4$ and $N = 50$, $n = 5$, $\|\boldsymbol{x}_0\|_0 = 2$. In this case, the estimates of base variables $\hat{x}_j$ are directly given as cube roots of the estimates of x_j^3. In both cases ($d = 2$ or 4), the success rates are similar to the ones reported in Tables 1 and 2, which provides evidence that the alternative proposed in Sect. 4 yields satisfactory results when Assumption 1 does not hold.

In order to compare with the greedy sparse-simplex (GSS) method of [Beck and Eldar, 2013] using the implementation provided with that paper, we consider a typical application for purely quadratic equations, namely, phase retrieval. In such applications, the measurements obey $c_i = |\boldsymbol{c}_i^T \boldsymbol{x}|$, which can be reformulated as $y_i = c_i^2 = \boldsymbol{x}^T \boldsymbol{c}_i \boldsymbol{c}_i^T \boldsymbol{x}$ to be handled by the proposed methods. Here, the components of the vectors $\boldsymbol{c}_i$, $i = 1, \ldots, N$, are randomly drawn in each trial according to a zero-mean Gaussian distribution of unit variance. Results in Table 5 obtained for $N = 25$, $n = 20$

and $\|\boldsymbol{x}_0\|_0 = 3$ show that the proposed methods perform much better than the GSS method with data generated in this manner, despite the fact that GSS works with the additional knowledge of $\|\boldsymbol{x}_0\|_0$. Note that many other methods have been proposed for sparse phase retrieval. The comparison is here limited to a sample of generic sparse recovery methods that apply to phase retrieval as a special case.

Remark: *All examples above consider a moderate dimensional setting ($n \leq 20$) with low sparsity levels $\|\boldsymbol{x}_0\|_0 \leq 3$. In such cases, the exact greedy algorithm (EGA) can be applied without complexity issues and yields a perfect recovery in all trials, even faster than the BP methods based on convex relaxations. However, despite its name and the observed success rates, the EGA is not an exact algorithm for solving polynomial systems but only for solving group-sparsity optimization problems in the form of (8). While Theorem 2 states that this is sufficient to solve (1)–(2) if the polynomial constraints are satisfied, there is no guarantee that this is the case in general.*

Estimating the probability of successful recovery. Figures 1 (for $n = 10$) and 2 (for $n = 20$) show the probability of successful recovery of an $\boldsymbol{x}_0$ estimated via the success rate versus the sparsity level $\|\boldsymbol{x}_0\|_0$ for various values of $\delta = N/n$. Contrary to the linear case where the system is overdetermined for values of $\delta > 1$, here δ can grow as large as M/n before the system becomes overdetermined. Thus, classical studies on the linear case focus on small values of $\delta < 1$, whereas we only analyze the probability of successful recovery for $\delta \geq 1$ ($\delta < 1$ is a very difficult setting with polynomial equations).

These results show that all the proposed methods can recover the sparsest solution for sufficiently sparse cases with high probability and that the sparsity level at which this occurs depends on the particular method on the one hand and on the ratio δ on the other hand. Larger values of δ correspond to larger systems with more equations and thus with more information on the sought solution. Another expected observation is that the probability of success decreases when the degree d increases. For $n = 20$, this leads to constant failure for the ℓ_1M and IR$\ell_1\ell_2$M methods at $d = 5$, though the S$\ell_1\ell_2$M and AGA methods can still recover sufficiently sparse solutions. Except for this particular setting ($n = 20$, $d = 5$), the results for IR$\ell_1\ell_2$M and S$\ell_1\ell_2$M are comparable, while the AGA yields slightly lower probabilities. The simple ℓ_1M method is much less effective than the others, but nonetheless obtains a high probability of recovery for not too difficult cases with $d = 2$ and a sufficiently large N.

Computational complexity. The next experiments evaluate the computational complexity of the methods with respect to the number of variables n, the degree d and the sparsity level $\|\boldsymbol{x}_0\|_0$. Focusing only on the computing time, we consider favorable settings with these parameters and the number of equations N set such that $\boldsymbol{x}_0$ should be recovered in most cases. The left plot of Fig. 3 shows the mean time of the methods for a range of $n \in [10, 40]$ with $\|\boldsymbol{x}_0\|_0 = 3$, $d = 2$ and $N = 50$. They all have a complexity exponential in n due to the dimension M of $\boldsymbol{\phi}$. In addition, they have similar rates of increase, except the EGA for which the computing time grows faster with n due to the combinatorial nature of the algorithm. Similar results, shown in the middle plot, are obtained with respect to the degree d (for $n = 10$, $\|\boldsymbol{x}_0\|_0 = 2$ and $N = 50$), except that the greedy algorithms benefit from a much lower rate of increase. In these experiments, the reweighting scheme of the S$\ell_1\ell_2$M is faster than the one used by the IR$\ell_1\ell_2$M, but its computing time highly depends on $\|\boldsymbol{x}_0\|_0$ whereas the number of iterations is fixed for IR$\ell_1\ell_2$M. This is clearly seen from the right plot of Fig. 3 (obtained with $n = 20$, $d = 2$ and $N = 100$), where the computing time of the S$\ell_1\ell_2$M reaches the one of the IR$\ell_1\ell_2$M when $\|\boldsymbol{x}_0\|_0$ equals the number of iterations of IR$\ell_1\ell_2$M. The exact greedy algorithm (EGA) suffers from a similar complexity issue with respect to $\|\boldsymbol{x}_0\|_0$, but much more pronounced. On the contrary, its approximate variant (AGA) remained very efficient in all of our tests.

6.2 Stable recovery in the presence of noise

We now turn to the noisy case as discussed in Sect. 5. In this setting, accuracy is measured by two performance indexes: the mean relative error, $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2/\|\boldsymbol{x}_0\|_2$, and the success rate with respect
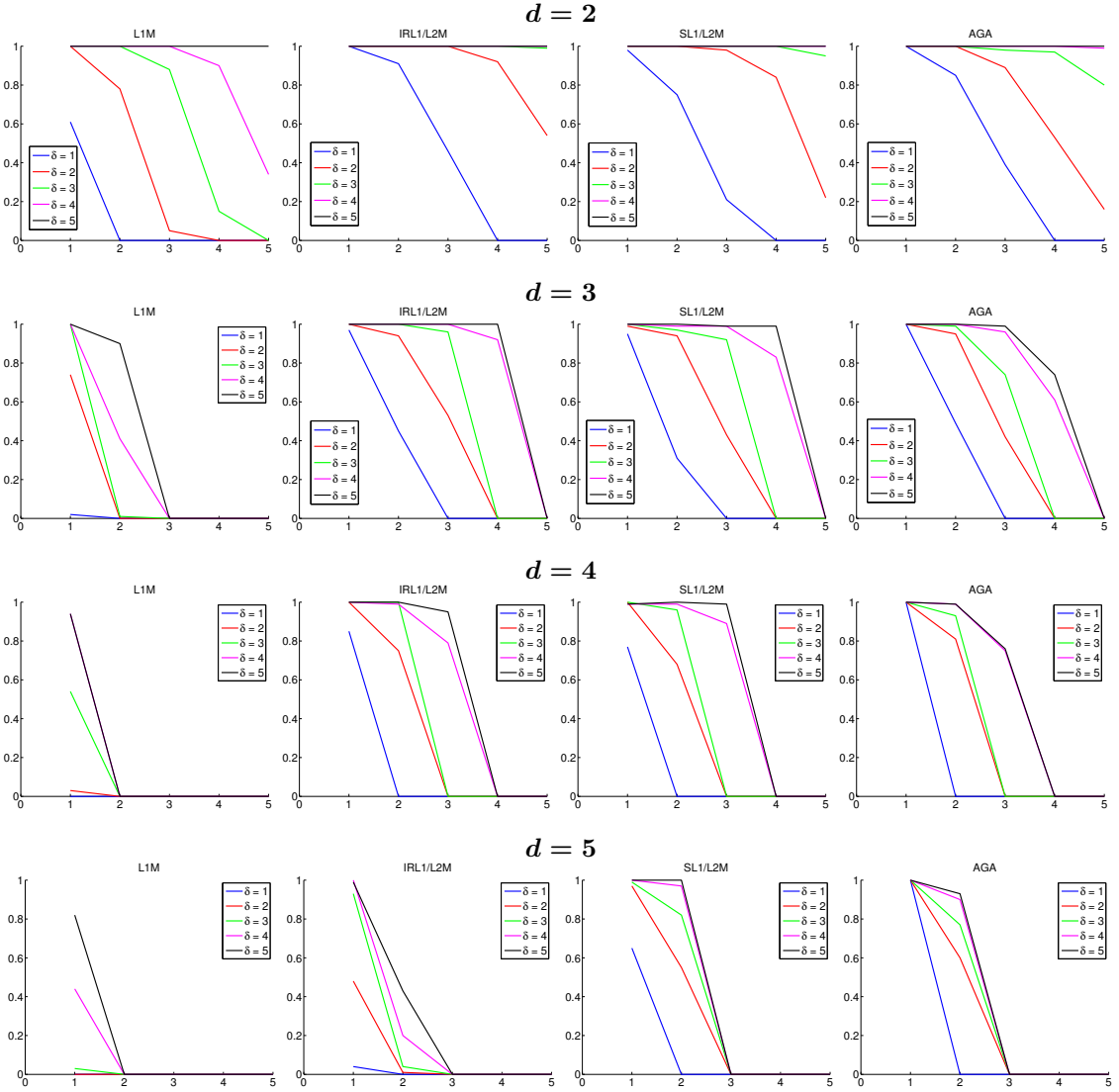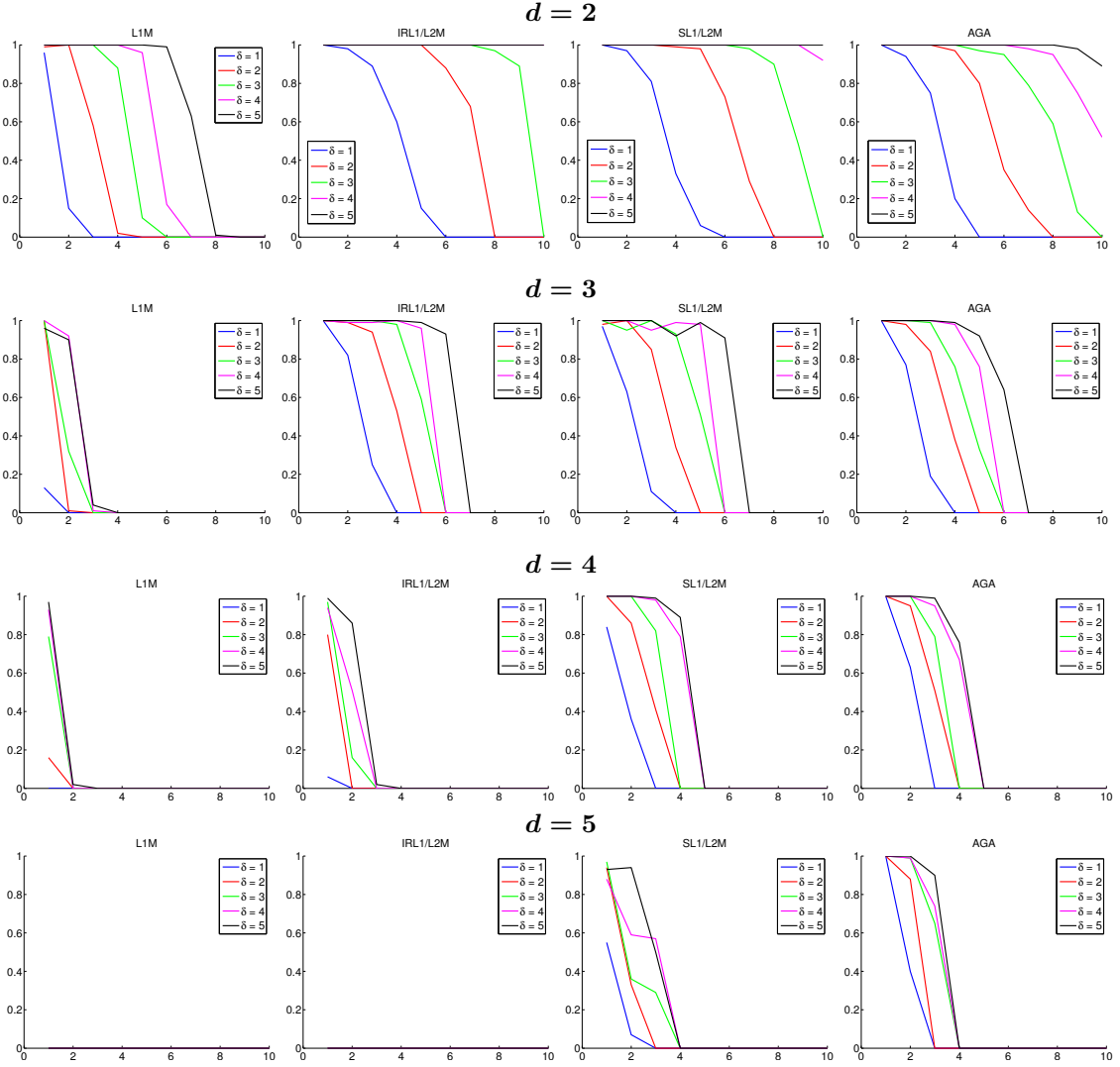
Figure 1: Estimated probability of successful recovery versus $\|\boldsymbol{x}_0\|_0$ for various $\delta = N/n$ in the case $n = 10$. Each row considers a different degree d, while each column contains the results of a different algorithm (from left to right: ℓ_1M, IR$\ell_1\ell_2$M, S$\ell_1\ell_2$M and AGA).

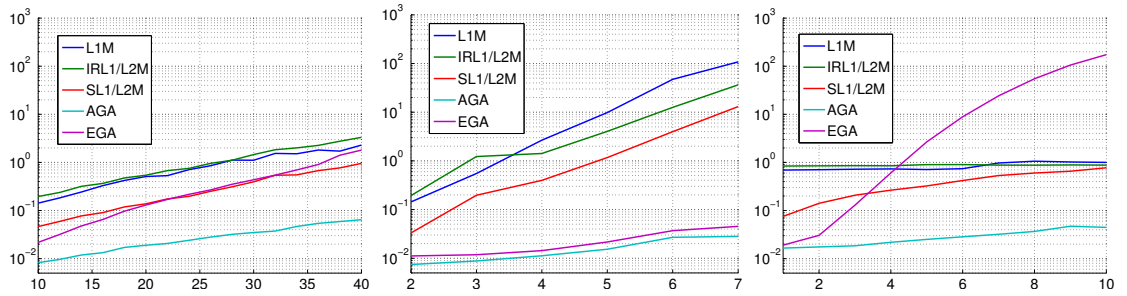Figure 2: Estimated probability of successful recovery versus $\|\boldsymbol{x}_0\|_0$ for various $\delta = N/n$ in the case $n = 20$. Each row considers a different degree d, while each column contains the results of a different algorithm (from left to right: ℓ_1M, IR$\ell_1\ell_2$M, S$\ell_1\ell_2$M and AGA).

Figure 3: Mean computing time of the methods in seconds versus the number of variables n (left), the degree d (middle) and the sparsity level $\|\boldsymbol{x}_0\|_0$ (right). These times are obtained on a Linux server equipped with 2 Xeon X5690 at 3.47 GHz.

Table 6: Results with noisy quadratic equations.

Method	QBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Mean relative error	16.1%	11.7%	22.2%	7.72%	6.52%	6.20%	6.19%
Success rate	14%	96%	0%	100%	100%	99%	100%
Mean time (s)	2.02	0.57	0.13	0.89	0.32	0.02	0.15

Table 7: Results with noisy equations of degree $d = 4$.

Method	NLBP	ℓ_1M	$\ell_1\ell_2$M	IR$\ell_1\ell_2$M	S$\ell_1\ell_2$M	AGA	EGA
Mean relative error	16.5%	22.3%	29.8%	7.65%	5.83%	6.74%	5.84%
Success rate	55%	87%	0%	100%	100%	99%	100%
Mean time (s)	13.6	0.31	0.12	0.75	0.19	0.006	0.007

to the estimation of the support of $\boldsymbol{x}_0$ (where values $|\hat{x}_j| < 10^{-6}$ are considered as zeros). Indeed, in many applications, the recovery of the correct support is more important that the precise values of the estimates.

First results. The data are generated as in Sect. 6.1 except that $y_i = p_i^d(\boldsymbol{x}_0) + e_i$, with noise terms e_i forming a zero-mean Gaussian random vector $\boldsymbol{e}$ satisfying $\|\boldsymbol{e}\|_2 = \varepsilon = 3$. This corresponds to a signal-to-noise ratio, $\|\boldsymbol{y}\|_2^2/\|\boldsymbol{e}\|_2^2$, of about 18 dB. Results with $N = 50$ polynomials of degree $d = 2$ in $n = 20$ variables are shown in Table 6 and in Table 7 for $d = 4$ and $n = 5$. The reweighted BP methods based on group-sparsity and the greedy methods lead to a rather small approximation error and they all recover the correct support in almost all trials despite the presence of noise.

Influence of the noise level ε. The influence of the noise level $\varepsilon = \|\boldsymbol{e}\|_2$ on the performance of the proposed methods is evaluated by letting the value of ε vary between 1 and 10, which corresponds to a signal-to-noise ratio decreased from 28 dB to 9 dB. For all methods except the ℓ_1M, the results plotted in Fig. 4 indicate that, as expected, e.g., from the bound (26), the approximation error directly depends on the noise level. In fact, the mean relative error is almost linear with respect to ε, which shows that the dependence on ε in the bound (26) of Theorem 6 is of the correct nature. But even more interestingly, the noise level does not influence the success rate corresponding to the estimation of the support of $\boldsymbol{x}_0$, thus providing evidence that the methods are robust to noise. Here again, the ℓ_1M method, which does not include structural knowledge in its formulation, does not benefit from such satisfactory features.
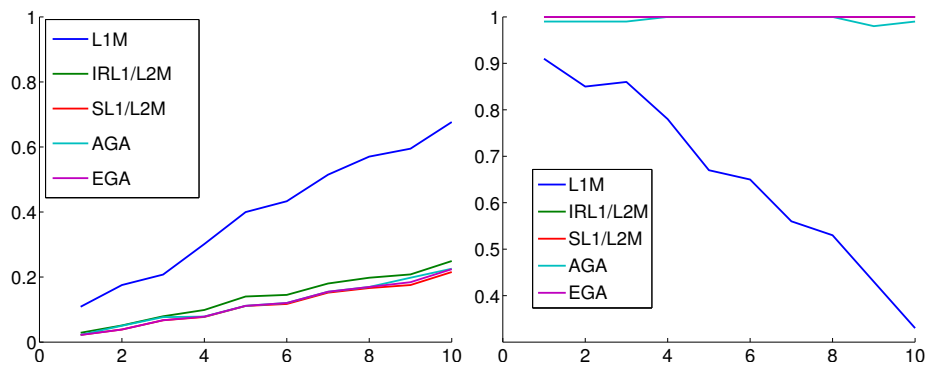

Figure 4: Mean relative error (left) and success rate (right) versus $\varepsilon = \|\boldsymbol{e}\|_2$. Except for the ℓ_1M, the curves are hardly distinguishable and close to 100% for the success rate.
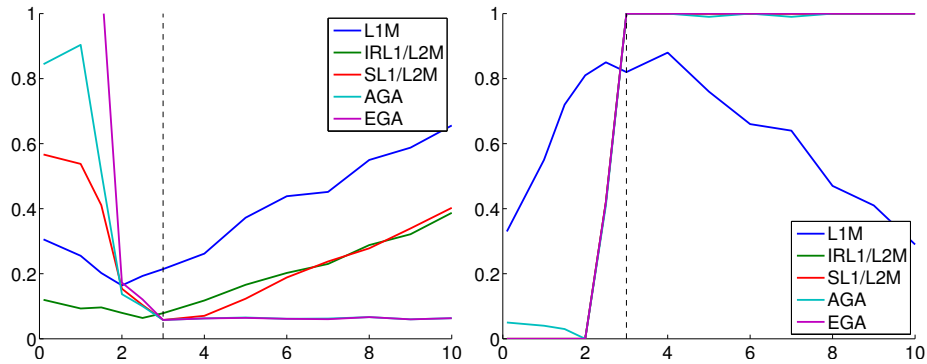
Figure 5: Mean relative error (left) and success rate (right) versus ε with data perturbed by a noise of ℓ_2-norm equal to 3. Except for ℓ_1M, the curves of the success rate are hardly distinguishable and close to 100% for $\varepsilon \geq 3$.

Influence of ε as a tuning parameter. In practical applications, the noise level might be unknown and ε becomes a tuning parameter. Figure 5 shows the influence of this parameter on the performance of the methods for a fixed noise level $\|\boldsymbol{e}\|_2 = 3$. All methods except the ℓ_1M perform very well with a slightly overestimated $\varepsilon \in [3, 4]$ and, for ε within a reasonable range around $\|\boldsymbol{e}\|_2$ ($\varepsilon \in [2, 6]$), the methods still yield rather accurate estimates of $\boldsymbol{x}_0$. Significantly larger values of ε lead to an increase of the error for all methods except the greedy algorithms which maintain a mean relative error below 7% for all values of $\varepsilon \geq \|\boldsymbol{e}\|_2$. On the other hand, the IR$\ell_1\ell_2$M method is the only one that is not badly affected by the underestimation of ε in terms of approximation error.

Regarding the estimation of the support of $\boldsymbol{x}_0$, all methods fail with $\varepsilon < \|\boldsymbol{e}\|_2$, while overestimating $\varepsilon \geq \|\boldsymbol{e}\|_2$ leads to perfect recovery even for much larger values of ε. This is in line with results on linear BP denoising, such as Theorem 4.1 in [Donoho et al., 2006] which guarantees that the estimated support is a subset of the sought one for sufficiently sparse cases when using an overestimated ε.

7 Conclusions

The paper proposed several methods for finding the sparsest solution of a system of polynomial equations. Two generic approaches were considered, one based on convex relaxations and one on a greedy strategy. For the convex relaxations, sufficient conditions of exact recovery of the sparsest solution were derived. The methods were also extended to deal with noisy equations, in which case stable recovery bounds for the convex relaxations were obtained. Both the computational efficiency and the accuracy of the proposed methods were shown in numerical experiments, which also emphasized the relationship between the probability of success and the sparsity of the solution for each method. In addition, these results indicate that the proposed methods accurately recover the sought solution in many cases where the sufficient conditions do not hold. As in classical BP theory, these conditions suffer from an "excessive pessimism" and are too restrictive due to their worst-case nature.

Remaining open issues include the convergence analysis of the greedy approximation towards the sparsest solution and the derivation of sufficient conditions for the solution of the group-sparsity optimization problem (8) to satisfy the polynomial constraints. Another question of interest is whether we can obtain less restrictive conditions for the variants of the convex relaxations with non-negativity constraints such as (10). Future work will also consider applying the proposed methods to more general nonlinear equations, e.g., by using Taylor expansions as in [Ohlsson et al., 2013a].

Acknowledgements

References

[Andersen and Andersen, 2000] Andersen, E. D. and Andersen, K. D. (2000). The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *High Performance Optimization*, 33:197–232.

[Balan et al., 2006] Balan, R., Casazza, P., and Edidin, D. (2006). On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, 20:345–356.

[Bandeira et al., 2014] Bandeira, A. S., Cahill, J., Mixon, D. G., and Nelson, A. A. (2014). Saving phase: Injectivity and stability for phase retrieval. *Applied and Computational Harmonic Analysis*, 37(1):106–125.

[Beck and Eldar, 2013] Beck, A. and Eldar, Y. C. (2013). Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509.

[Blumensath, 2013] Blumensath, T. (2013). Compressed sensing with nonlinear observations and related nonlinear optimisation problems. *IEEE Transactions on Information Theory*, 59(6):3466–3474.

[Blumensath and Davies, 2008] Blumensath, T. and Davies, M. E. (2008). Gradient pursuit for non-linear sparse signal modelling. In *European Signal Processing Conference (EUSIPCO)*, pages 25–29.

[Bruckstein et al., 2009] Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81.

[Candès, 2006] Candès, E. J. (2006). Compressive sampling. In *Proceedings of the International Congress of Mathematicians: invited lectures*, pages 1433–1452.

[Candès et al., 2013a] Candès, E. J., Eldar, Y. C., Strohmer, T., and Voroninski, V. (2013a). Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225.

[Candès et al., 2013b] Candès, E. J., Strohmer, T., and Voroninski, V. (2013b). Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274.

[Candès et al., 2008] Candès, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905.

[Deng et al., 2011] Deng, W., Yin, W., and Zhang, Y. (2011). Group sparse optimization by alternating direction method. Technical Report TR11-06, Department of Computational and Applied Mathematics, Rice University.

[Donoho, 2006] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

[Donoho et al., 2006] Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18.

[Donoho and Huo, 2001] Donoho, D. L. and Huo, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862.

[Ehler et al., 2014] Ehler, M., Fornasier, M., and Sigl, J. (2014). Quasi-linear compressed sensing. *Multiscale Modeling & Simulation*, 12(2):725–754.

[Eldar and Kutyniok, 2012] Eldar, Y. C. and Kutyniok, G., editors (2012). *Compressed Sensing: Theory and Applications*. Cambridge University Press.

[Fienup, 1982] Fienup, J. (1982). Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769.

[Foucart and Rauhut, 2013] Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Springer.

[Gerchberg and Saxton, 1972] Gerchberg, R. and Saxton, W. (1972). A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246.

[Gonsalves, 1976] Gonsalves, R. (1976). Phase retrieval from modulus data. *Journal of Optical Society of America*, 66(9):961–964.

[Grant and Boyd, 2008] Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited. http://stanford.edu/~boyd/graph_dcp.html.

[Grant and Boyd, 2013] Grant, M. and Boyd, S. (2013). CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx.

[Kohler and Mandel, 1973] Kohler, D. and Mandel, L. (1973). Source reconstruction from the modulus of the correlation function: a practical approach to the phase problem of optical coherence theory. *Journal of the Optical Society of America*, 63(2):126–134.

[Le et al., 2013] Le, V. L., Lauer, F., and Bloch, G. (2013). Selective ℓ_1 minimization for sparse recovery. *IEEE Transactions on Automatic Control*. (to appear).

[Marchesini, 2007] Marchesini, S. (2007). Phase retrieval and saddle-point optimization. *Journal of the Optical Society of America A*, 24(10):3289–3296.

[Ohlsson and Eldar, 2013] Ohlsson, H. and Eldar, Y. C. (2013). On conditions for uniqueness in sparse phase retrieval. *CoRR*, abs/1308.5447.

[Ohlsson et al., 2013a] Ohlsson, H., Yang, A. Y., Dong, R., and Sastry, S. (2013a). Nonlinear basis pursuit. In *Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, USA*, pages 315–319.

[Ohlsson et al., 2013b] Ohlsson, H., Yang, A. Y., Dong, R., Verhaegen, M., and Sastry, S. (2013b). Quadratic basis pursuit. *arXiv preprint arXiv:1301.7002*.

[Ranieri et al., 2013] Ranieri, J., Chebira, A., Lu, Y. M., and Vetterli, M. (2013). Phase retrieval for sparse signals: Uniqueness conditions. *CoRR*, abs/1308.3058.

[Vidal et al., 2005] Vidal, R., Ma, Y., and Sastry, S. (2005). Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959.

A Bound on the sparsity level of ϕ

Lemma 3. *For all $(a, b, d) \in (\mathbb{N}^*)^3$ such that $a \geq d(b + d)$, the inequality*

$$\frac{1}{a} \sum_{q=2}^{d} \binom{a+q-1}{q} \geq \frac{d}{b} \sum_{q=2}^{d} \binom{b+q-1}{q}$$

holds.

Proof. For $q \leq d$, we can bound the terms in the sum as

$$\frac{1}{a} \binom{a+q-1}{q} = \frac{(a+q-1)!}{a\, q!(a-1)!} = \frac{1}{a\, q!} \prod_{i=0}^{q-1}(a+i) = \frac{1}{q!} \prod_{i=1}^{q-1}(a+i)$$

$$\geq \frac{1}{q!} a^{q-1}$$

$$\geq \frac{1}{q!} d^{q-1}(b+d)^{q-1}$$

$$\geq \frac{1}{q!} d^{q-1} \prod_{i=1}^{q-1}(b+i)$$

$$\geq \frac{d^{q-1}}{b} \binom{b+q-1}{q}$$

where we used $a \geq d(b+d)$ in the second inequality. Then

$$\frac{1}{a} \sum_{q=2}^{d} \binom{a+q-1}{q} \geq \frac{1}{b} \sum_{q=2}^{d} d^{q-1} \binom{b+q-1}{q}$$

$$\geq \frac{d}{b} \sum_{q=2}^{d} d^{q-2} \binom{b+q-1}{q}$$

$$\geq \frac{d}{b} \sum_{q=2}^{d} \binom{b+q-1}{q}$$

$\square$

Proposition 2. *Let the mapping $\phi : \mathbb{R}^n \to \mathbb{R}^M$ be defined as above. Then, with $d \geq 3$ and $n \geq d(\|\boldsymbol{x}\|_0 + d)$, the vector $\phi(\boldsymbol{x})$ is sparser than the vector $\boldsymbol{x}$ in the sense that the inequality*

$$\frac{\|\phi(\boldsymbol{x})\|_0}{M} \leq \frac{2\|\boldsymbol{x}\|_0}{dn}$$

holds for all $\boldsymbol{x} \in \mathbb{R}^n$.

Proof. By construction, the number of nonzeros in $\phi(\boldsymbol{x})$ is equal to the sum over q, $1 \leq q \leq d$, of the number of monomials of degree q in $\|\boldsymbol{x}\|_0$ variables:

$$\frac{\|\phi(\boldsymbol{x})\|_0}{M} = \frac{\sum_{q=1}^{d} \binom{\|\boldsymbol{x}\|_0 + q - 1}{q}}{\sum_{q=1}^{d} \binom{n+q-1}{q}} = \frac{\|\boldsymbol{x}\|_0}{n} \frac{\frac{1}{\|\boldsymbol{x}\|_0} \sum_{q=1}^{d} \binom{\|\boldsymbol{x}\|_0 + q - 1}{q}}{\frac{1}{n} \sum_{q=1}^{d} \binom{n+q-1}{q}}$$

$$= \frac{\|\boldsymbol{x}\|_0}{n} \frac{1 + \frac{1}{\|\boldsymbol{x}\|_0} \sum_{q=2}^{d} \binom{\|\boldsymbol{x}\|_0 + q - 1}{q}}{1 + \frac{1}{n} \sum_{q=2}^{d} \binom{n+q-1}{q}}$$

$$= \frac{\|\boldsymbol{x}\|_0}{n} \left[\frac{1}{1 + \frac{1}{n} \sum_{q=2}^{d} \binom{n+q-1}{q}} + \frac{\frac{1}{\|\boldsymbol{x}\|_0} \sum_{q=2}^{d} \binom{\|\boldsymbol{x}\|_0 + q - 1}{q}}{1 + \frac{1}{n} \sum_{q=2}^{d} \binom{n+q-1}{q}} \right]$$

The assumption $n \geq d(\|\boldsymbol{x}\|_0 + d)$ implies that[6] $d \leq n$. With $d \leq n$, we have

$$\frac{1}{n}\binom{n+q-1}{q} \geq \frac{1}{q!}n^{q-1} \geq \frac{1}{q!}d^{q-1}$$

which yields

$$\frac{1}{n}\sum_{q=2}^{d}\binom{n+q-1}{q} \geq \frac{1}{n}\sum_{q=2}^{d}\binom{n+2-1}{2} \geq (d-1)\frac{d}{2}$$

Now, on the one hand we have

$$d \geq 3 \Rightarrow \frac{1}{2}d^2 - \frac{3}{2}d + 1 \geq 0$$

$$\Rightarrow 1 + \frac{1}{n}\sum_{q=2}^{d}\binom{n+q-1}{q} \geq d$$

$$\Rightarrow \frac{1}{1 + \frac{1}{n}\sum_{q=2}^{d}\binom{n+q-1}{q}} \leq \frac{1}{d}$$

and on the other hand, Lemma 3 yields

$$\frac{\frac{1}{\|\boldsymbol{x}\|_0}\sum_{q=2}^{d}\binom{\|\boldsymbol{x}\|_0+q-1}{q}}{1 + \frac{1}{n}\sum_{q=2}^{d}\binom{n+q-1}{q}} \leq \frac{\frac{1}{\|\boldsymbol{x}\|_0}\sum_{q=2}^{d}\binom{\|\boldsymbol{x}\|_0+q-1}{q}}{\frac{1}{n}\sum_{q=2}^{d}\binom{n+q-1}{q}} \leq \frac{1}{d}$$

Thus,

$$\frac{\|\phi(\boldsymbol{x})\|_0}{M} \leq \frac{2\|\boldsymbol{x}\|_0}{dn}$$

$\square$

B Other conditions for sparse recovery

The following uses the exact value of $\|\boldsymbol{\phi}_0\|_0$.

Theorem 7. *Let $\boldsymbol{x}_0$ denote the unique solution to (1)–(2). If the inequality*

$$\sum_{q=1}^{d}\binom{\|\boldsymbol{x}_0\|_0+q-1}{q} \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right) \tag{33}$$

holds, then the solution $\hat{\boldsymbol{\phi}}$ to (5) is unique and equal to $\phi(\boldsymbol{x}_0)$, thus providing $\hat{\boldsymbol{x}} = \boldsymbol{x}_0$.

Another more compact but slightly less tight result is as follows.

Theorem 8. *Let $\boldsymbol{x}_0$ denote the unique solution to (1)–(2). If the inequality*

$$\binom{\|\boldsymbol{x}_0\|_0+d-1}{d} \leq \frac{1}{2d}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right) \tag{34}$$

holds, then the solution $\hat{\boldsymbol{\phi}}$ to (5) is unique and equal to $\phi(\boldsymbol{x}_0)$, thus providing $\hat{\boldsymbol{x}} = \boldsymbol{x}_0$.

Proof. Since the terms in the sum of Theorem 7 form an increasing sequence, we have

$$\sum_{q=1}^{d}\binom{\|\boldsymbol{x}_0\|_0+q-1}{q} \leq d\binom{\|\boldsymbol{x}_0\|_0+d-1}{d}$$

which yields the sought statement by application of Theorem 7.

$\square$

---

[6]Assume $d > n$, then, the assumption of the Proposition leads to $n > n(\|\boldsymbol{x}\|_0 + n)$ and $\|\boldsymbol{x}\|_0 + n < 1$ which is impossible since $n \geq 1$.

C   Value of m

Let us define m as the number of monomials involving a base variable x with a degree ≥ 1. It can be computed as the sum over q, $0 \leq q \leq d - 1$ of the number of monomials of degree q in $n - 1$ variables times the remaining degree $d - q$ (since each monomial in $n - 1$ variables can be multiplied by x or x^2... or x^{d-q} to produce a monomial of degree at most d in n variables):

$$m = \sum_{q=0}^{d-1}(d - q)\binom{(n-1)+q-1}{q} = \sum_{q=0}^{d-1}(d-q)\binom{n+q-2}{q}.$$

Another technique computes m as the total number of all monomials minus the number of monomials not involving x which is the number of monomials in $n - 1$ variables:

$$
\begin{aligned}
m &= M - \sum_{q=1}^{d}\binom{n+q-2}{q} \\
&= \sum_{q=1}^{d}\binom{n+q-1}{q} - \binom{n+q-2}{q} \\
&= \sum_{q=1}^{d}\left[\frac{(n+q-1)}{n-1} - 1\right]\binom{n+q-2}{q} \\
&= \sum_{q=1}^{d}\frac{q}{n-1}\binom{n+q-2}{q}.
\end{aligned}
$$