



HAL
open science

L2-L0 regularization path tracking algorithms

Charles Soussen, Jérôme Idier, Junbo Duan, David Brie

► **To cite this version:**

Charles Soussen, Jérôme Idier, Junbo Duan, David Brie. L2-L0 regularization path tracking algorithms. 2014. hal-00948313v1

HAL Id: hal-00948313

<https://hal.science/hal-00948313v1>

Submitted on 18 Feb 2014 (v1), last revised 18 Mar 2015 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ℓ_2 - ℓ_0 regularization path tracking algorithms

Charles Soussen*, Jérôme Idier, Junbo Duan, and David Brie

Abstract

Sparse signal approximation can be formulated as the mixed ℓ_2 - ℓ_0 minimization problem $\min_{\mathbf{x}} \mathcal{J}(\mathbf{x}; \lambda) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0$. We propose two heuristic search algorithms to minimize \mathcal{J} for a continuum of λ -values, yielding a sequence of coarse to fine approximations. Continuation Single Best Replacement is a bidirectional greedy algorithm adapted from the Single Best Replacement algorithm previously proposed for minimizing \mathcal{J} for fixed λ . ℓ_0 regularization path track is a more complex algorithm exploiting that the ℓ_2 - ℓ_0 regularization path is piecewise constant with respect to λ . Tracking the ℓ_0 regularization path is done in a sub-optimal manner by maintaining (i) a list of subsets that are candidates to be solution supports for decreasing λ 's and (ii) the list of critical λ -values around which the solution changes. Both algorithms gradually construct the ℓ_0 regularization path by performing single replacements, *i.e.*, adding or removing a dictionary atom from a subset. A straightforward adaptation of these algorithms yields sub-optimal solutions to $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ subject to $\|\mathbf{x}\|_0 \leq k$ for contiguous values of $k \geq 0$ and to $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \varepsilon$ for continuous values of ε . Numerical simulations show the effectiveness of the algorithms on a difficult sparse deconvolution problem inducing a highly correlated dictionary \mathbf{A} .

Index Terms

Sparse signal estimation; ℓ_0 -constrained least-squares; ℓ_0 -penalized least-squares; ℓ_2 - ℓ_0 regularization path; stepwise algorithms; Orthogonal Least Squares; continuation.

C. Soussen and D. Brie are with the Université de Lorraine and the CNRS at the Centre de Recherche en Automatique de Nancy (CRAN, UMR 7039). Campus Sciences, B.P. 70239, F-54506 Vandœuvre-lès-Nancy, France. Tel: (+33)-3 83 59 56 43, Fax: (+33)-3 83 68 44 62. E-mail: charles.soussen@univ-lorraine.fr, david.brie@univ-lorraine.fr.

J. Idier is with L'UNAM Université, Ecole Centrale Nantes and the CNRS at the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN, UMR 6597), 1 rue de la Noë, BP 92101, F-44321 Nantes Cedex 3, France. Tel: (+33)-2 40 37 69 09, Fax: (+33)-2 40 37 69 30. E-mail: jerome.idier@irccyn.ec-nantes.fr.

J. Duan was with CRAN. He now is with the Department of Biomedical Engineering, Xi'an Jiaotong University. No. 28, Xianning West Road, Xi'an 710049, Shaanxi Province, China. Tel: (+86)-29-82 66 86 68, Fax: (+86)-29 82 66 76 67. E-mail: junbo.duan@mail.xjtu.edu.cn.

I. INTRODUCTION

Sparse approximation from noisy data is traditionally addressed as the constrained least-square problems

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k \quad (1)$$

or

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \varepsilon \quad (2)$$

where $\|\mathbf{x}\|_0$ is the ℓ_0 -“norm” counting the number of nonzero entries in \mathbf{x} , and the quadratic fidelity-to-data term $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ measures the quality of approximation. Formulation (1) is well adapted when one has a knowledge of the maximum number k of atoms to be selected in the dictionary \mathbf{A} . On the contrary, it may arise that k is unknown but one has a knowledge of the variance of the observation noise, leading to the choice of (2) with an appropriate value of ε , related to the noise variance. Since both (1) and (2) are subset selection problems, they are discrete optimization problems. They are known to be NP-hard except for specific cases [1].

When no knowledge is available on either k and ε , the unconstrained formulation

$$\min_{\mathbf{x}} \{\mathcal{J}(\mathbf{x}; \lambda) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0\} \quad (3)$$

is worth being considered, where λ expresses the trade-off between the quality of approximation and the sparsity level [2]. In a Bayesian viewpoint, (3) can be seen as a limit maximum *a posteriori* formulation where $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ and the penalty $\|\mathbf{x}\|_0$ are respectively related to a Gaussian noise distribution and a prior distribution for sparse signals (specifically, a limit Bernoulli-Gaussian distribution with infinite Gaussian variance) [3]. Moreover, (3) is well suited to the design of forward-backward (also called “bidirectional”) algorithms that update the support of \mathbf{x} by adding or removing a dictionary atom at each iteration. Indeed, they can be naturally interpreted as descent algorithms to minimize $\mathcal{J}(\mathbf{x}; \lambda)$ [3], [4].

A. Classification of methods

1) *ℓ_0 -constrained least-squares*: Let us first consider the constrained least-square problems (1) and (2) for *fixed* k or ε . The dedicated discrete optimization algorithms can be categorized into two classes. First, the forward greedy algorithms explore subsets of increasing cardinalities starting from the empty set. At each iteration, a new atom is appended to the current subset, therefore gradually refining the approximation [5]. Greedy algorithms include, by increasing order of complexity: Matching Pursuit (MP) [6], Orthogonal Matching Pursuit (OMP) [7], and Orthogonal Least Squares (OLS) [8], also

referred to as forward selection in statistical regression [9] and known as Order Recursive Matching Pursuit (ORMP) [10] and Optimized Orthogonal Matching Pursuit (OOMP) [11]. The second category of discrete algorithms dedicated to (1) are thresholding algorithms, where each iteration delivers a subset of *same* cardinality k . Popular thresholding algorithms include Iterative Hard Thresholding [12], [13], Subspace Pursuit [14] and CoSaMP [15], [16].

Among these two categories, greedy algorithms are specifically well-adapted to the resolution of (1) and (2) for *variable* sparsity levels k . Indeed, they yield a series of subsets for consecutive cardinalities k (*i.e.*, for decreasing approximation errors ε) since at each iteration, the current subset is increased by one element.

2) *ℓ_0 -penalized least-squares*: In [3], we evidenced that the minimization of $\mathcal{J}(\mathbf{x}; \lambda)$ using a descent algorithm naturally leads to bidirectional extensions of forward (orthogonal) greedy algorithms. To be more specific, consider the ℓ_0 constrained least-square problem (1) and a given selected support \mathcal{Q} . It is clear that the inclusion of a new element into \mathcal{Q} yields a decrease of the least squared error $\mathcal{E}_{\mathcal{Q}}$, defined as the minimum of $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ for \mathbf{x} supported by \mathcal{Q} . Conversely, an atom de-selection increases the approximation error. Thus, a descent algorithm dedicated to (1) takes the form of a forward strategy where no atom de-selection is allowed. On the contrary, the ℓ_0 -penalized cost function $\mathcal{J}(\mathbf{x}; \lambda)$ may decrease with both an atom selection or de-selection. Therefore, formulation (3) allows one to design a descent scheme based on a bidirectional search strategy. The algorithms Single Best Replacement (SBR) [3] and Bayesian OMP [4] have been proposed in this spirit. They are bidirectional descent algorithms adapted from OLS and OMP, respectively, for ℓ_0 -penalized least-square minimization. At each iteration, they modify the current subset by one element. This *single replacement* consists in appending or removing an atom from the current subset. The advantage of bidirectional algorithms over forward greedy algorithms is that an early wrong atom selection may be later cancelled. Bidirectional algorithms include the so-called stepwise regression algorithms which are OLS forward-backward extensions [9], [17], [18], and OMP based forward-backward extensions of lower complexity [4], [19].

In this paper, we will address the ℓ_0 -penalized formulation for various (continuous) sparsity levels λ and propose new sub-optimal algorithms. The set of solutions to (3) for all λ -values will be referred to as the ℓ_0 regularization path.

3) *Connection with the continuous relaxation of the ℓ_0 norm*: The algorithms described so far are essentially discrete search algorithms to solve the problems (1), (2) or (3) involving the ℓ_0 norm. A popular alternative approach relies on (i) the relaxation of the ℓ_0 norm by a continuous function, convex or not, that is nondifferentiable at 0; and (ii) the continuous optimization of the resulting cost function.

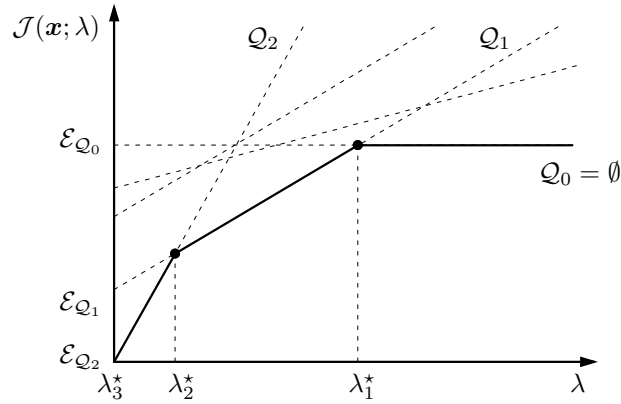


Fig. 1. Representation of lines $\lambda \mapsto \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|$ for various subsets \mathcal{Q} . The “ ℓ_0 curve”, in plain line, is the minimal curve $\lambda \mapsto \min_{\mathcal{Q}} \{\mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|\}$. It is continuous, concave, and piecewise affine with a finite number of pieces. It characterizes the ℓ_0 regularization path.

See, *e.g.*, [20], [21] for ℓ_1 minimization and [22]–[27] for nonconvex optimization. It is noticeable that the ℓ_1 -norm relaxation leads to algorithms yielding sparse approximations for consecutive cardinalities [20], [28]. In particular, the well-known homotopy algorithm recovers the ℓ_1 regularization path. It reads as a bidirectional greedy algorithm whose complexity is close to that of OMP [28], [29]. This algorithm will be considered in the simulation section for comparison purposes (see Section V).

B. Two main ideas

The first and main idea developed here is dedicated to ℓ_0 -penalized least-squares for various λ -values. It allows us to design heuristic search strategies for tracking the ℓ_0 regularization path. The second idea is a straightforward adaptation to address (1) and (2) for various values of k and ε .

1) *Approach for ℓ_0 -penalized least-squares:* The cost function $\mathcal{J}(\mathbf{x}; \lambda)$ handles the trade-off between low residual $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ and low cardinality $\|\mathbf{x}\|_0$. Our approach is based on the following geometric interpretation.

First, any subset \mathcal{Q} yields a (set of) least-square solution \mathbf{x} supported by \mathcal{Q} . The cost $\mathcal{J}(\mathbf{x}_{\mathcal{Q}}; \lambda)$ associated to the solutions $\mathbf{x}_{\mathcal{Q}}$ having the sparsest supports is represented by the line of equation $\lambda \mapsto \mathcal{E}_{\mathcal{Q}} + \lambda\|\mathbf{x}_{\mathcal{Q}}\|_0$ (see Fig. 1) where $\mathcal{E}_{\mathcal{Q}} \triangleq \|\mathbf{y} - \mathbf{A}\mathbf{x}_{\mathcal{Q}}\|_2^2$ stands for the least-square error.

Second, the ℓ_0 regularization path is piecewise constant with respect to λ (see Appendix A for a rigorous proof). Geometrically, this result can be easily understood by noticing that the minimum value of $\mathcal{J}(\mathbf{x}; \lambda)$ with respect to \mathbf{x} is obtained for all λ -values by considering the concave envelope of the set

of lines $\lambda \mapsto \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|$ for all subsets \mathcal{Q} , where $|\mathcal{Q}|$ denotes the cardinality operator¹. The resulting piecewise affine curve will be referred to as the “ ℓ_0 curve” (see Fig. 1). Its edges are related to the best sparse approximation supports for all λ , and its vertices are the critical λ -values around which the set of optimal solutions $\arg \min_{\mathbf{x}} \mathcal{J}(\mathbf{x}; \lambda)$ is changing.

We take advantage of this geometric interpretation to propose two suboptimal search algorithms, named “Continuation Single Best Replacement” (CSBR) and “ ℓ_0 regularization path track” (ℓ_0 -PT) to address (3) for a continuum of λ -values. CSBR repeatedly minimizes $\mathcal{J}(\mathbf{x}; \lambda)$ with respect to \mathbf{x} for decreasing λ values. It is a greedy bidirectional search where the current subset is locally modified at any iteration: all the possible single replacements are tested. ℓ_0 -PT is a more complex search maintaining a list of candidate subsets (for CSBR, only the current subset is updated), each corresponding to an edge of the ℓ_0 curve. Local searches are performed from subsets in the list so as to update the current evaluation of the ℓ_0 curve. Both algorithms yield sparse approximations for continuous sparsity levels λ that are adaptively delivered by the algorithm.

2) *Approach for ℓ_0 -constrained least-squares:* We propose a straightforward adaptation of both algorithms to address (1) and (2) for consecutive values of k or continuous ε . The adaptation simply amounts to storing the “best subset” explored by the tracking algorithm for any cardinality, *i.e.*, the explored subset of cardinality k yielding the least squared error.

C. Related works

1) *Connection with bi-objective optimization:* The tracking problem introduced above can be linked to the bi-objective optimization literature [30]. The formulations (1), (2) and (3) are related to the same bi-objective optimization problem because they all intend to minimize both the approximation error $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ and the sparsity measure $\|\mathbf{x}\|_0$. Although \mathbf{x} is continuous valued, the bi-objective optimization problem should rather be considered as a discrete one where both objectives reread $\mathcal{E}_{\mathcal{Q}}$ and $|\mathcal{Q}|$. Indeed, there is a one-to-one correspondence between the solutions \mathbf{x} and \mathcal{Q} of both problems, $\mathbf{x} = \mathbf{x}_{\mathcal{Q}}$ reading as the least-square minimizer on support \mathcal{Q} (²).

¹When a line is “minimal”, it is easy to see that $\|\mathbf{x}_{\mathcal{Q}}\|_0 = |\mathcal{Q}|$, *i.e.*, all least-squares coefficients x_i are non-zero since otherwise, \mathcal{Q} could be reduced leading to a new line laying below the line related to \mathcal{Q} . Thus, we now consider the lines $\lambda \mapsto \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|$ instead of $\lambda \mapsto \mathcal{E}_{\mathcal{Q}} + \lambda\|\mathbf{x}_{\mathcal{Q}}\|_0$.

²When the subdictionary $\mathbf{A}_{\mathcal{Q}}$ indexed by \mathcal{Q} is not full column rank, there are several least-square minimizers. However, when \mathbf{x} is a global minimizer of (3) for some λ -value, the support \mathcal{Q} of \mathbf{x} yields a full rank matrix $\mathbf{A}_{\mathcal{Q}}$ [2].

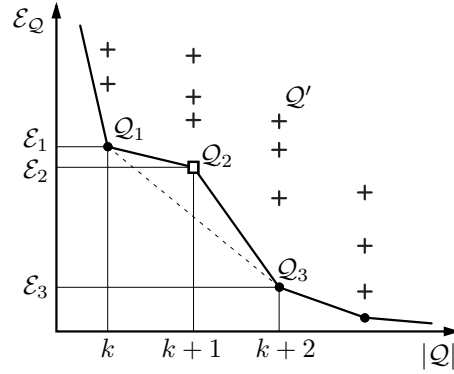


Fig. 2. Sparse approximation seen as a bi-objective optimization problem. The Pareto frontier gathers the *non-dominated* points: no other point can strictly decrease both $|Q|$ and \mathcal{E}_Q . Bullets and squares are all non-dominated points whereas ‘+’ denotes dominated points. A *supported* solution is a minimizer of $\mathcal{E}_Q + \lambda|Q|$ with respect to Q for some λ : see the representation of Fig. 1 in the plane (λ, \mathcal{J}) . Q_1 and Q_3 are supported contrary to Q_2 .

Fig. 2 is a classical bi-objective representation where each axis is related to a single objective [31], namely $|Q|$ and \mathcal{E}_Q . In bi-objective optimization, a point Q is called Pareto optimal when no other point Q' can decrease both objectives [30]. In the present context, $|Q|$ takes integer values, thus the Pareto solutions are obviously the minimizers over Q of \mathcal{E}_Q subject to $|Q| \leq k$ for some value of k . Equivalently, they minimize $|Q|$ subject to $\mathcal{E}_Q \leq \varepsilon$ for some ε . The Pareto frontier gathers the Pareto solutions, *i.e.*, the optimal solutions to both (1) for all k and (2) for all ε . The Pareto solutions are usually classified as supported and non-supported efficient solutions. The former lay in the convex envelope of the Pareto frontier (the bullet points in Fig. 2) whereas the latter lay in the nonconvex areas (the square point). It is well known that any supported solution can be reached by the weighted sum method, *i.e.*, when minimizing $\mathcal{E}_Q + \lambda|Q|$ with respect to Q for some λ -value, while the non-supported solutions cannot [30].

2) ℓ_0 regularization path tracking seen as a weighted sum method: In multi-objective optimization, the weighted sum method is usually considered as a way to transform a difficult optimization problem with multiple constraints into a simpler unconstrained mono-objective problem. However, the non-supported solutions cannot be reached using the weighted sum method when some objectives are not convex. Specifically, the weighted sum formulation (3) may not yield the same solutions as the ℓ_0 constrained formulations (1) and (2) because the ℓ_0 norm is nonconvex [2]. Choosing between the weighting sum method and a more complex method delivering non-supported solutions is a nontrivial question. The answer depends on the problem at-hand and specifically, on the size of the nonconvex areas in the Pareto

frontier.

We point out that the previous discussion assumes that an optimal algorithm is available for the weighted sum method. In our context, the minimization of $\mathcal{J}(\mathbf{x}; \lambda)$ is acknowledged to be difficult because \mathcal{J} may have a very large number of local minimizers. In the recent sparse approximation literature, many authors actually discourage the direct optimization of \mathcal{J} for this reason [22], [24]. In [3], however, we showed that OLS bidirectional extensions are able to “escape” from some local minimizers of $\mathcal{J}(\mathbf{x}; \lambda)$ for a given sparsity level λ . This motivates us to propose efficient OLS based solutions for minimizing \mathcal{J} for variable λ -values.

3) *Positioning with respect to other stepwise algorithms:* In statistical regression, the word “stepwise” originally refers to Efroymsen’s algorithm [17], proposed in 1960 as an empirical extension of forward selection (*i.e.*, OLS). Other stepwise algorithms were proposed in the 1980’s [9, Chapter 3] among which Berk’s and Broersen’s algorithms [18], [32]. All these algorithms perform a single replacement per iteration and were originally applied to over-determined problems in which the number of columns of \mathbf{A} is lower than the number of rows. More recent forward-backward algorithms were designed as either OMP [4], [19] or OLS extensions [33], [34]. Their common feature is that they aim to find subsets of cardinality k yielding a low residual $\mathcal{E}_{\mathcal{Q}}$ for all k . Although our algorithms share the same objective, they are based on the refinement of the ℓ_0 regularization path. To the best of our knowledge, the idea of tracking the ℓ_0 regularization path is novel. Moreover, we design descent algorithms to minimize $\mathcal{J}(\mathbf{x}; \lambda)$ for a continuum of λ -values while most stepwise algorithms are empirical variations of OLS without any obvious connection with the cost function $\mathcal{J}(\mathbf{x}; \lambda)$.

4) *Connection with the Single Best Replacement algorithm:* In [3], we proposed the SBR algorithm to address (3) for a *specific* sparsity level λ . It is an OLS forward-backward extension in which at each iteration, the single replacement yielding the largest decrease of $\mathcal{J}(\cdot; \lambda)$ is selected. Contrary to SBR, the proposed CSBR and ℓ_0 -PT algorithms deliver sub-optimal solutions for a continuum of λ -values. They yield subsets of increasing cardinalities, each being associated to an interval of λ -values in such a way that the resulting intervals partition \mathbb{R}_+ . SBR, CSBR and ℓ_0 -PT all read as descent algorithms in different senses. SBR minimizes the cost $\mathcal{J}(\cdot; \lambda)$ for a specific λ whereas CSBR minimizes $\mathcal{J}(\cdot; \lambda)$ for decreasing λ -values by repeatedly calling SBR. Finally, ℓ_0 -PT minimizes $\mathcal{J}(\cdot; \lambda)$ for any λ -value simultaneously (an iteration does not match a specific λ anymore). Although CSBR and ℓ_0 -PT both perform a series of single replacements from candidate subsets, ℓ_0 -PT is not a direct extension of SBR. Since our first proposal of CSBR in the conference paper [35], we realized that CSBR can be enhanced by adopting single replacement rules that differ from the SBR rules. This led us to elaborate the ℓ_0 -PT version. Also,

the underlying idea of ℓ_0 -PT, namely tracking the ℓ_0 curve was not developed in [35]. Nevertheless, the structure of ℓ_0 -PT is more complex than that of CSBR. Because practical users may be interested in simple (yet efficient) algorithms, and because CSBR outperforms SBR which is already acknowledged as an efficient algorithm in the community [27], we feel that CSBR is worth being presented in the present paper as well.

5) *Positioning with respect to continuation algorithms:* Generally speaking, the principle of continuation is to handle a difficult problem by solving a sequence of simpler problems with warm start initialization, and gradually tuning some hyperparameter [36]. In sparse approximation, the continuation terminology often refers to relaxation methods replacing the ℓ_0 -norm by the ℓ_1 -norm. The resulting ℓ_2 - ℓ_1 optimization problem is solved for decreasing values of the hyperparameter using the solution for each value as the starting point for the next value³ [5]. In particular, the homotopy algorithm [28], [29], [39] takes into account that the ℓ_2 - ℓ_1 regularization path is piecewise affine, and tracks the critical hyperparameter values characterizing the changes in the solution support, *i.e.*, the changepoints between two consecutive affine intervals. CSBR is designed in a similar manner (although the ℓ_2 - ℓ_0 minimization steps are solved in a sub-optimal way) by repeatedly calling SBR for decreasing λ -values that are recursively computed. On the contrary, ℓ_0 -PT may be seen as a continuation algorithm in some weak sense only. Although sparse solutions are delivered for continuous hyperparameter values λ , ℓ_0 -PT does not rely on sequential resolutions of (3) with decreasing λ 's. It is rather a fully discrete approach that gradually improves the estimated ℓ_0 regularization path.

The paper is organized as follows. In Section II, we properly define the ℓ_2 - ℓ_0 regularization path and establish its main properties. In Section III, we propose the CSBR algorithm extending SBR for a continuum of decreasing λ -values. In Section IV, the ℓ_0 -PT algorithm is proposed based on the piecewise constant property of the (optimal) ℓ_2 - ℓ_0 regularization path. Although sub-optimal, ℓ_0 -PT also reconstructs a piecewise constant path. In Section V, both proposed algorithms are analyzed on a difficult sparse deconvolution problem. We show that the recovered ℓ_0 regularization paths are more accurate than the ℓ_1 regularization path obtained by homotopy, and that the performance of OLS and SBR are outperformed as well. Finally, we investigate the automatic choice of the cardinality k using classical order selection rules.

³Note that in [37], the word “continuation” has a totally different meaning. It is used to denote a Graduated Non Convexity (GNC) like approach close to that of [38], where the ℓ_0 pseudo-norm is relaxed by a series of concave metrics leading to the resolution of a series of continuous optimization problems with warm start initialization.

II. OPTIMAL ℓ_2 - ℓ_0 REGULARIZATION PATHS

A. Basic definitions and working assumptions

Let $m \times n$ denote the size of the dictionary \mathbf{A} (usually, $m \leq n$ in sparse approximation). The observation signal \mathbf{y} and the weight vector \mathbf{x} are thus of size $m \times 1$ and $n \times 1$, respectively. We assume that any $\min(m, n)$ columns of \mathbf{A} are linearly independent so that for any subset \mathcal{Q} , the submatrix of \mathbf{A} gathering the columns indexed by \mathcal{Q} is full rank, and the least-square error $\mathcal{E}_{\mathcal{Q}}$ can be numerically computed. This assumption is however not necessary for the theoretical results provided in the appendix. On the algorithmic viewpoint, the full rank assumption may be relaxed provided that there exists a simple way to check that a set of columns are linearly independent. Similar to [3], the proposed algorithm can be straightforwardly adapted by forbidding to explore any subset associated to linearly dependent columns.

Given a subset $\mathcal{Q} \subset \{1, \dots, n\}$ of cardinality lower than $\min(m, n)$, we recall that $\mathbf{x}_{\mathcal{Q}}$ denotes the related least squares solution and $\mathcal{E}_{\mathcal{Q}} = \|\mathbf{y} - \mathbf{A}\mathbf{x}_{\mathcal{Q}}\|_2^2$. Obviously, we have $\|\mathbf{x}_{\mathcal{Q}}\|_0 \leq |\mathcal{Q}|$. In [3], we showed that in non trivial cases involving noisy data, $\|\mathbf{x}_{\mathcal{Q}}\|_0 = |\mathcal{Q}|$ almost surely, *i.e.*, all entries in $\mathbf{x}_{\mathcal{Q}}$ are non-zero.

B. Definition and properties of the ℓ_2 - ℓ_0 regularization path

We now properly define the ℓ_2 - ℓ_0 regularization path and state its main properties. The piecewise constant property (Theorem 1) is the starting point of the tracking algorithm presented in Section IV.

For $k \leq \min(m, n)$, let $\mathcal{X}_c(k)$ denote the set of solutions to the ℓ_0 -constrained least-square problem:

$$\mathcal{X}_c(k) = \arg \min_{\mathcal{Q}} \mathcal{E}_{\mathcal{Q}} \quad \text{subject to} \quad |\mathcal{Q}| \leq k. \quad (4)$$

In the same way, for $\lambda > 0$, let $\mathcal{X}_p(\lambda)$ gather the ℓ_0 -penalized least-square minimizers:

$$\mathcal{X}_p(\lambda) = \arg \min_{\mathcal{Q}} \{\mathcal{J}_{\mathcal{Q}}(\lambda) \triangleq \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|\} \quad (5)$$

with the implicit constraint $|\mathcal{Q}| \leq \min(m, n)$. By extension, let also $\mathcal{X}_p(+\infty) = \{\emptyset\}$.

Theorem 1 $\mathcal{X}_p(\lambda)$ is a piecewise constant function of λ : there exists a decreasing sequence $\lambda_0^* \triangleq +\infty > \lambda_1^* > \dots > \lambda_I^* > \lambda_{I+1}^* \triangleq 0$ such that $\mathcal{X}_p(\lambda)$ is constant on each interval $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$.

Proof: See Appendix A. ■

λ_i^* will be referred to as the *critical values* (see Fig. 1). A direct consequence of Theorem 1 is that the ℓ_0 curve is piecewise affine since all curves $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda)$ are affine. We can now properly define the notion of regularization path.

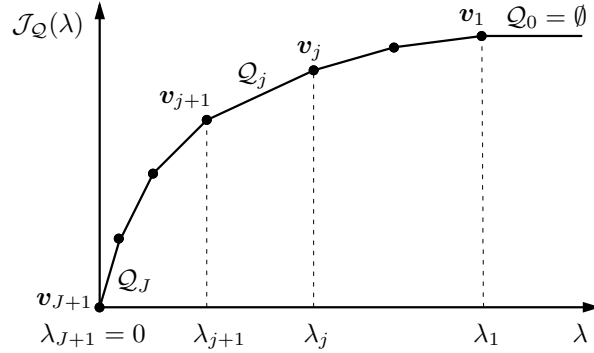


Fig. 3. The estimated ℓ_0 regularization path is parametrized by subsets \mathcal{Q}_j (with $\mathcal{Q}_0 = \emptyset$) and the critical λ -values λ_j around which the solution changes. The ℓ_0 curve is described by the 2D vertices \mathbf{v}_j : the edge linking \mathbf{v}_j and \mathbf{v}_{j+1} is supported by the line $\lambda \mapsto \mathcal{J}_{\mathcal{Q}_j}(\lambda)$.

Definition 1 The constrained regularization path is the finite set (of sets) $\mathcal{X}_c = \{\mathcal{X}_c(k), k = 0, \dots, \min(m, n)\}$. Similarly, the ℓ_2 - ℓ_0 regularization path is defined as $\mathcal{X}_p = \{\mathcal{X}_p(\lambda), \lambda > 0\}$. \mathcal{X}_p contains a finite number of sets $\mathcal{X}_p(\lambda)$ according to Theorem 1.

The regularization paths \mathcal{X}_c and \mathcal{X}_p may not coincide. This is actually a consequence of the non convexity of the ℓ_0 -norm [2], [31]. Specifically, $\mathcal{X}_p \subset \mathcal{X}_c$ as stated in Theorem 2 below, but the proposition “ $\mathcal{X}_c(k) \subset \mathcal{X}_p$ ” may be false. For the example of Fig. 2, one can easily check that $\mathcal{Q}_2 \in \mathcal{X}_c(k+1)$ but $\mathcal{Q}_2 \notin \mathcal{X}_p$ since for any λ , $\mathcal{J}_{\mathcal{Q}_2}(\lambda) > \min(\mathcal{J}_{\mathcal{Q}_1}(\lambda), \mathcal{J}_{\mathcal{Q}_3}(\lambda))$.

Theorem 2 $\mathcal{X}_p \subset \mathcal{X}_c$ and for any $\lambda \notin \{\lambda_1^*, \dots, \lambda_0^*\}$, there exists k such that $\mathcal{X}_p(\lambda) = \mathcal{X}_c(k)$.

Proof: See Appendix A. ■

C. Parametrization of the estimated ℓ_0 regularization path

Because the optimal ℓ_2 - ℓ_0 regularization path is piecewise constant, we impose that our tracking algorithms always yield a piecewise constant (sub-optimal) regularization path, and that the related ℓ_0 curve is a piecewise affine function with identical endpoints. Let us now introduce some notations.

Similar to the definition of the critical values λ_i^* for the optimal ℓ_2 - ℓ_0 regularization path, let λ_j refer to the estimated regularization path. $\{\lambda_j, j = 0, \dots, J+1\}$ is a decreasing sequence with $\lambda_0 \triangleq +\infty$ and $\lambda_{J+1} \triangleq 0$ (see Fig. 3). Additionally, let \mathcal{Q}_j denote the sub-optimal solution to (5) for $\lambda \in (\lambda_{j+1}, \lambda_j)$, and by extension, $\mathcal{Q}_0 \triangleq \emptyset$ for $\lambda > \lambda_1$. Using these notations, the ℓ_0 curve is the 2D path $\mathbf{v} = \{\mathbf{v}_1, \dots, \mathbf{v}_{J+1}\}$ in

the (λ, \mathcal{J}) domain, with $\mathbf{v}_j = (\lambda_j, \mathcal{E}_{\mathcal{Q}_j} + \lambda_j |\mathcal{Q}_j|)$. Our ℓ_0 regularization path tracking procedures estimate the critical values λ_j and the corresponding subsets \mathcal{Q}_j by gradually refining the ℓ_0 regularization path.

III. GREEDY CONTINUATION ALGORITHM (CSBR)

Our starting point is the Single Best Replacement algorithm [3] dedicated to the minimization of $\mathcal{J}(\mathbf{x}; \lambda)$ with respect to \mathbf{x} , or equivalently to $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda |\mathcal{Q}|$ with respect to \mathcal{Q} . We first briefly recall the SBR algorithm for a given λ . Then, the CSBR extension is presented for decreasing and adaptive λ -values.

A. Single Best Replacement

SBR is a deterministic descent algorithm dedicated to the minimization of $\mathcal{J}_{\mathcal{Q}}(\lambda)$ with the initial solution $\mathcal{Q} = \emptyset$. Let us denote by \mathcal{Q} the active subset and by $\mathcal{Q} \bullet i$ a single replacement, *i.e.*, the insertion or removal of a dictionary column i into/from the active set \mathcal{Q} :

$$\mathcal{Q} \bullet i \triangleq \begin{cases} \mathcal{Q} \cup \{i\} & \text{if } i \notin \mathcal{Q}, \\ \mathcal{Q} \setminus \{i\} & \text{otherwise.} \end{cases} \quad (6)$$

An SBR iteration is based on (i) the computation of $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$ for all i (n insertion and removal trials) and (ii) the selection of the replacement $\mathcal{Q} \bullet \ell$ yielding the minimal value of $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$:

$$\ell \in \arg \min_{i \in \{1, \dots, n\}} \mathcal{J}_{\mathcal{Q} \bullet i}(\lambda). \quad (7)$$

SBR terminates when no replacement decreases the cost function. The stopping condition $\forall i, \mathcal{J}_{\mathcal{Q} \bullet i}(\lambda) \geq \mathcal{J}_{\mathcal{Q}}(\lambda)$ rereads:

$$\underbrace{\max_{i \notin \mathcal{Q}} \{\mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q} \cup \{i\}}\}}_{\lambda^+} \leq \lambda \leq \underbrace{\min_{i \in \mathcal{Q}} \{\mathcal{E}_{\mathcal{Q} \setminus \{i\}} - \mathcal{E}_{\mathcal{Q}}\}}_{\lambda^-}. \quad (8)$$

When $\lambda > 0$, SBR terminates after a finite number of iterations because it is a descent algorithm and there are a finite number of possibilities for the active set $\mathcal{Q} \subset \{1, \dots, n\}$. In the limit case $\lambda = 0$, we have $\mathcal{J}_{\mathcal{Q}}(0) = \mathcal{E}_{\mathcal{Q}}$. Only insertions are performed since any removal increases the squared error $\mathcal{E}_{\mathcal{Q}}$. SBR thus coincides with the well known OLS algorithm [8]. Generally, the n replacement trials necessitate to compute $\mathcal{E}_{\mathcal{Q} \bullet i}$ for all i . In [3], we proposed an efficient (fast and stable) recursive implementation based on the Cholesky factorization of the Gram matrix $\mathbf{A}_{\mathcal{Q}}^t \mathbf{A}_{\mathcal{Q}}$ when \mathcal{Q} is modified by one element (where $\mathbf{A}_{\mathcal{Q}}$ stands for the submatrix of \mathbf{A} gathering the active columns).

SBR is summarized in Tab. I (in the standard version, the lines within brackets are omitted) and illustrated in Fig. 4(a). Geometrically, a single replacement yields to a vertical displacement (from top

TABLE I

SBR ALGORITHM FOR MINIMIZATION OF $\mathcal{J}_{\mathcal{Q}}(\lambda)$ WITH RESPECT TO \mathcal{Q} FOR FIXED λ . IN THE STANDARD VERSION [3], $\mathcal{Q}_{\text{init}} = \emptyset$ AND THE BOLD LINES WITHIN BRACKETS ARE OMITTED. IN BOLD, WE MENTION THE SMALL ADAPTATIONS WHEN SBR IS REPEATEDLY CALLED BY CSBR (SEE TAB. II). THE BEST SBR ITERATES ARE DENOTED BY $\mathcal{Q}_k^{\text{CSBR}}$ AND THE RELATED LEAST SQUARED ERRORS BY $\mathcal{E}_k^{\text{CSBR}}$.

Inputs: \mathbf{A} , \mathbf{y} , λ , active set $\mathcal{Q}_{\text{init}}$
 $[i_{\text{init}} \notin \mathcal{Q}_{\text{init}}$ and tables $\mathcal{Q}_k^{\text{CSBR}}$ and $\mathcal{E}_k^{\text{CSBR}}$]

Step 0: Set **iter** = 1 and $\mathcal{Q} = \mathcal{Q}_{\text{init}}$.
[Set iter = 2 and $\mathcal{Q} = \mathcal{Q}_{\text{init}} \cup \{i_{\text{init}}\}$]

Step 1: For $i \in \{1, \dots, n\}$, compute $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$.
[If iter = 2]
[Compute $\ell \in \arg \min_{i \neq i_{\text{init}}} \mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$]
[Else]
 Compute $\ell \in \arg \min_i \mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$.
[End if]

If $\mathcal{J}_{\mathcal{Q} \bullet \ell}(\lambda) < \mathcal{J}_{\mathcal{Q}}(\lambda)$,
 Set $\mathcal{Q} = \mathcal{Q} \bullet \ell$.
 Else,
 Terminate SBR.
[Compute λ^+ and i^+ according to (9) and (10)]

End if.
 Set **iter** = **iter** + 1 and go to Step 1.

Outputs: • $\mathcal{Q} = \text{SBR}(\mathcal{Q}_{\text{init}}; \lambda)$.
 $[[\mathcal{Q}, \lambda^+, i^+] = \text{SBR}(\mathcal{Q}_{\text{init}}; \lambda, i_{\text{init}})]$ [Updated tables $\mathcal{Q}_k^{\text{CSBR}}$ and $\mathcal{E}_k^{\text{CSBR}}$]

to bottom) between the lines $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda)$ and $\lambda \mapsto \mathcal{J}_{\mathcal{Q} \bullet \ell}(\lambda)$ associated to consecutive active sets. By default, the initial active set is empty [3]. In the following, we propose a continuation strategy based on recursive calls to SBR for decreasing λ -values (from infinity to 0) with the last SBR output as initial solution. In subsection III-B, we propose a recursive solution to decrease λ adaptively to the data. The proposed CSBR algorithm is finally detailed in subsection III-C.

B. Principle of the continuation algorithm

Consider the execution of SBR for a given $\lambda = \lambda_j$ yielding the support $\mathcal{Q} = \text{SBR}(\mathcal{Q}_{\text{init}}; \lambda_j)$ as output, where $\mathcal{Q}_{\text{init}}$ stands for the initial support. The stopping condition (8) is thus fulfilled for active set \mathcal{Q} and $\lambda = \lambda_j$. Moreover, the output of $\text{SBR}(\mathcal{Q}; \lambda)$ is equal to \mathcal{Q} whenever $\lambda < \lambda_j$ is larger than λ^+ defined

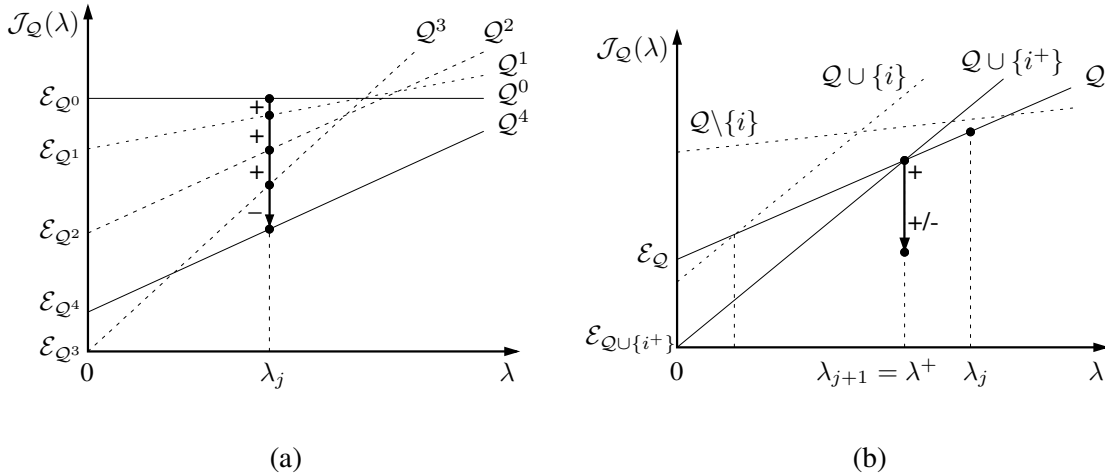


Fig. 4. Graphical interpretation of SBR and CSBR. (a) SBR: for a specific $\lambda = \lambda_j$, a single replacement $\mathcal{Q} \bullet \ell$ yields a vertical displacement (from top to bottom) from lines $\mathcal{J}_{\mathcal{Q}}(\lambda)$ to $\mathcal{J}_{\mathcal{Q} \bullet \ell}(\lambda)$. The slope $|\mathcal{Q}|$ is increased (insertion) or decreased (removal) by one. The initial active set is empty (horizontal line). Here, three insertions and a removal are being performed. (b) CSBR: recursive computation of λ . \mathcal{Q} is the output of SBR for $\lambda = \lambda_j$: all dashed lines $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$ lay above $\mathcal{J}_{\mathcal{Q}}(\lambda)$ when $\lambda = \lambda_j$. The line associated to $\mathcal{Q} \cup \{i^+\}$ is located below all other parallel lines $\mathcal{Q} \cup \{i\}$. i^+ is the first index to be appended into \mathcal{Q} during the call $\text{SBR}(\mathcal{Q}; \lambda_{j+1})$.

in (8) since (8) is fulfilled again. Therefore, we propose to perform the next call to $\text{SBR}(\mathcal{Q}; \lambda_{j+1})$ with

$$\lambda_{j+1} = \lambda^+ = \lambda_j + \mathcal{J}_{\mathcal{Q}}(\lambda_j) - \min_{i \notin \mathcal{Q}} \mathcal{J}_{\mathcal{Q} \cup \{i\}}(\lambda_j). \quad (9)$$

When $\lambda < \lambda^+$, the stopping condition (8) is violated, hence the cost $\mathcal{J}_{\mathcal{Q}}(\lambda)$ decreases with a single replacement (insertion). For $\lambda = \lambda^+$, the first inequality in (8) becomes an equality: $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{J}_{\mathcal{Q} \cup \{i\}}(\lambda)$ for some value of i , named i^+ . To avoid any confusion regarding the non-strict decrease of the cost function, the SBR execution for $\lambda = \lambda^+$ shall be understood as the limit case of the behavior of SBR for $\lambda \rightarrow \lambda^+$, $\lambda < \lambda^+$. In other words, we impose that in the first iteration of $\text{SBR}(\mathcal{Q}; \lambda_{j+1})$, an atom indexed by:

$$i^+ \in \arg \max_{i \notin \mathcal{Q}} \{\mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q} \cup \{i\}}\} = \arg \min_{i \notin \mathcal{Q}} \mathcal{J}_{\mathcal{Q} \cup \{i\}}(\lambda_j) \quad (10)$$

is selected. In the second iteration, the single replacement tests are all performed except the removal of i^+ to avoid infinite loops. This small adaptation of SBR is summarized (bold lines within brackets) in Tab. I, together with the computation of λ^+ and i^+ , now provided as algorithm outputs. Because all values of $\mathcal{J}_{\mathcal{Q} \cup \{i\}}(\lambda_j)$ are computed during the insertion trials, λ^+ can be directly computed from (9) with almost no additional cost at the last iteration of SBR. Fig. 4(b) illustrates that:

TABLE II

CSBR ALGORITHM. THE BEST EXPLORED SUBSETS ARE DENOTED BY $\mathcal{Q}_k^{\text{CSBR}}$ AND THE SBR OUTPUTS BY \mathcal{Q}_j .

Inputs: \mathbf{A} , \mathbf{y} , K and/or ε
Set $j = 1$, $\lambda_0 = +\infty$, $\mathcal{Q}_0 = \emptyset$.
Compute λ_1 and i^+ using (11).
While ($\lambda_j > 0$), ($ \mathcal{Q}_{j-1} < K$) and ($\mathcal{E}_{\mathcal{Q}_{j-1}} > \varepsilon$),
Call $[\mathcal{Q}_j, \lambda_{j+1}, i^+] = \text{SBR}(\mathcal{Q}_{j-1}; \lambda_j, i^+)$.
Do $j = j + 1$.
End while
Outputs: • Best subsets $\mathcal{Q}_k^{\text{CSBR}}$ and related squared errors $\mathcal{E}_k^{\text{CSBR}}$ ($k = 0, \dots, K$).
• Estimated regularization path: lists of critical values λ_j , list of subsets \mathcal{Q}_j and the squared errors $\mathcal{E}_{\mathcal{Q}_j}$.

- for $\lambda < \lambda_j$, any removal increases the cost function $\mathcal{J}_{\mathcal{Q}}(\lambda)$: the dashed line related to $\mathcal{Q} \setminus \{i\}$ lays above the one related to \mathcal{Q} ;
- similarly, for $\lambda \geq \lambda_j$, any insertion $\mathcal{Q} \cup \{i\}$ increases the cost function $\mathcal{J}_{\mathcal{Q}}(\lambda)$;
- the lines \mathcal{Q} and $\mathcal{Q} \cup \{i^+\}$ intersect for $\lambda = \lambda^+$. The + label in the figure refers to the selection of i^+ at the first iteration of $\text{SBR}(\mathcal{Q}; \lambda^+)$. The +/- arrow represents further single replacements occurring from the second iteration.

C. CSBR algorithm

The structure of CSBR is summarized in Tab. II. The calls $\mathcal{Q}_j = \text{SBR}(\mathcal{Q}_{j-1}; \lambda_j)$ deliver subsets for decreasing λ_j with $\lambda_0 = +\infty$ and $\mathcal{Q}_0 = \emptyset$. \mathcal{Q}_j is the sub-optimal solution corresponding to all λ -values in $(\lambda_{j+1}, \lambda_j]$, and the λ_j values are updated according to (9) with $\mathcal{Q} \leftarrow \mathcal{Q}_j$. At the very first iteration, we have $\mathcal{Q}_0 = \emptyset$, and (10) yields:

$$i^+ \in \arg \max_{i \in \{1, \dots, n\}} \frac{|\langle \mathbf{y}, \mathbf{a}_i \rangle|}{\|\mathbf{a}_i\|_2} \quad \text{and} \quad \lambda_1 = \frac{\langle \mathbf{y}, \mathbf{a}_{i^+} \rangle^2}{\|\mathbf{a}_{i^+}\|_2^2}. \quad (11)$$

The CSBR stopping conditions involve a maximum cardinality ($|\mathcal{Q}_j| \geq K$) and/or a threshold on the squared error ($\mathcal{E}_{\mathcal{Q}_j} \leq \varepsilon$). Additionally, CSBR stops when $\lambda_{j+1} \leq 0$, which means that the whole range of sparsity levels $\lambda \in (0, +\infty)$ has been scanned. This condition is however rarely met when dealing with real noisy data. In any case, SBR is never stopped before convergence (here, recall that SBR terminates after a finite number of iterations). In the pseudo-code version of Tab. II, CSBR yields k -sparse approximations for consecutive k 's up to the storage of the best intermediate SBR iterates. A subset \mathcal{Q} is stored as the ‘‘best iterate’’ of cardinality $k = |\mathcal{Q}|$ if the squared error $\mathcal{E}_{\mathcal{Q}}$ is lower than that of

the already stored iterate of same cardinality k . The sequence of best SBR iterates is updated whenever SBR is called (see Tab. I). This yields sparse supports for contiguous $k \in \{0, \dots, K\}$ because the initial support is empty and a series of single replacements are performed to explore nested subsets.

Note that a given support can never be explored twice while running SBR because SBR is a descent algorithm for fixed λ . On the contrary, a support might be explored twice using CSBR (during two calls to SBR for different λ -values) but never indefinitely. As illustrated in Fig. 4(b), each λ_j -value is associated with the intersection between two lines $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda)$ and $\lambda \mapsto \mathcal{J}_{\mathcal{Q} \cup \{i\}}(\lambda)$. Because there are a finite number of such 2D lines, the number of possible intersections is finite.

IV. TRACKING THE ℓ_0 REGULARIZATION PATH (ℓ_0 -PT)

As seen in Section II, the optimal ℓ_2 - ℓ_0 regularization path is characterized by a polygonal and concave ℓ_0 curve (Fig. 1). Here, we propose to gradually refine some concave ℓ_0 curve (represented in Fig. 3) by updating the list of critical values $\{\lambda_1, \dots, \lambda_J\}$ and the corresponding subsets $\{\mathcal{Q}_0, \dots, \mathcal{Q}_J\}$. For CSBR, the iteration number j identifies with the j -th subset \mathcal{Q}_j in the path because the path is gradually constructed by working for decreasing λ_j values. Therefore, the interval $(\lambda_{j+1}, \lambda_j]$ found in the j -th iteration is never updated in the subsequent iterations. On the contrary, at each iteration of the ℓ_0 -PT algorithm, the ℓ_0 regularization path is already constructed on $\lambda \in (0, +\infty)$. ℓ_0 -PT performs a path *refinement*: some subset \mathcal{Q}_j is selected and a local search is performed to improve the ℓ_0 regularization path. When an improvement occurs, the refined ℓ_0 curve lays below the former. Let us now specify how the support \mathcal{Q}_j is selected (subsection IV-A) and explored (subsection IV-B).

A. Selection of the support \mathcal{Q}_j to be explored

Assume that some current estimation of the ℓ_0 regularization path is available according to the concave representation of Fig. 3. For any subset \mathcal{Q}_j in the path, let us define the Boolean indicator $\text{explored}(j) = 1$ if \mathcal{Q}_j has already been explored in the previous iterations (*i.e.*, the path improvements induced by single replacements from \mathcal{Q}_j have already been taken into account), and $\text{explored}(j) = 0$ otherwise. The current iteration of ℓ_0 -PT selects the unexplored support \mathcal{Q}_j of lowest cardinality. Therefore, $\mathcal{Q}_0, \dots, \mathcal{Q}_{j-1}$ have necessarily been already explored: indeed, the cardinality of \mathcal{Q}_j increases with j by concavity of the estimated ℓ_0 curve (Fig. 3).

B. Exploration of support \mathcal{Q}_j

Once $\mathcal{Q} = \mathcal{Q}_j$ has been selected, ℓ_0 -PT attempts to modify it by performing single replacements:

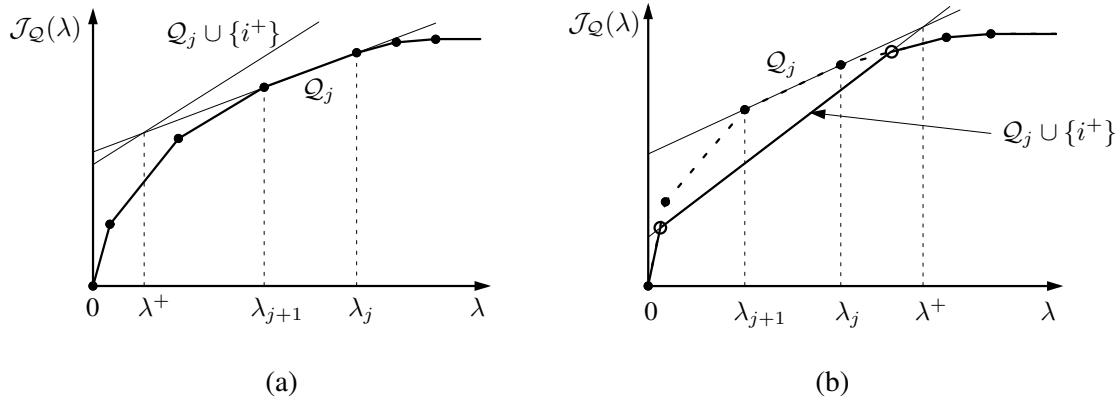


Fig. 5. Best insertion into the subset \mathcal{Q}_j belonging to the ℓ_0 regularization path. (a) When $\lambda^+ \leq \lambda_{j+1}$, the new line $\mathcal{Q}_j \cup \{i^+\}$ lays above the ℓ_0 curve, which is thus not improved. (b) When $\lambda^+ > \lambda_{j+1}$, $\mathcal{Q}_j \cup \{i^+\}$ intersects one or two edges of the ℓ_0 curve. $\mathcal{Q}_j \cup \{i^+\}$ is inserted as a new edge and all edges laying above the line $\mathcal{Q}_j \cup \{i^+\}$ are removed.

- Test all possible insertions $\mathcal{Q} \cup \{i\}$ and attempt to include the best subset $\mathcal{Q} \cup \{i^+\}$ in the regularization path.
- Test all possible removals $\mathcal{Q} \setminus \{i\}$ and attempt to include the best subset $\mathcal{Q} \setminus \{i^-\}$ in the regularization path.
- If \mathcal{Q} still belongs to the ℓ_0 curve, mark it as explored.

In particular, when no single replacement can improve the path, \mathcal{Q}_j is labeled as explored and the path is unchanged.

1) *Insertion tests:* All possible insertions $\mathcal{Q} \cup \{i\}$ ($i \notin \mathcal{Q}$) are tested by computing the squared errors $\mathcal{E}_{\mathcal{Q} \cup \{i\}}$. Similar to SBR and CSBR, this task amounts to solving $n - |\mathcal{Q}|$ least-square problems. The best insertion i^+ is given in (10). Geometrically, both lines $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}|$ and $\mathcal{J}_{\mathcal{Q} \cup \{i^+\}}(\lambda) = \mathcal{E}_{\mathcal{Q} \cup \{i^+\}} + \lambda(|\mathcal{Q}| + 1)$ intersect at $\lambda = \lambda^+$. Moreover, if $\lambda^+ \leq \lambda_{j+1}$, the latter lays above the ℓ_0 curve by concavity of the ℓ_0 curve; see Fig. 5(a). Thus, no improvement of the ℓ_0 curve is possible. When $\lambda^+ > \lambda_{j+1}$, there are one or two intersections between $\mathcal{J}_{\mathcal{Q} \cup \{i^+\}}(\lambda)$ and the ℓ_0 -curve. ℓ_0 -PT updates it by inserting $\mathcal{Q} \cup \{i^+\}$ as a new edge and removing all existing edges laying above it (see Fig. 5(b)).

2) *Removal tests:* We adopt a similar analysis. The removal yielding the least squared error is given by:

$$i^- = \arg \min_{i \in \mathcal{Q}} \{\mathcal{E}_{\mathcal{Q} \setminus \{i\}} - \mathcal{E}_{\mathcal{Q}}\} \quad (12)$$

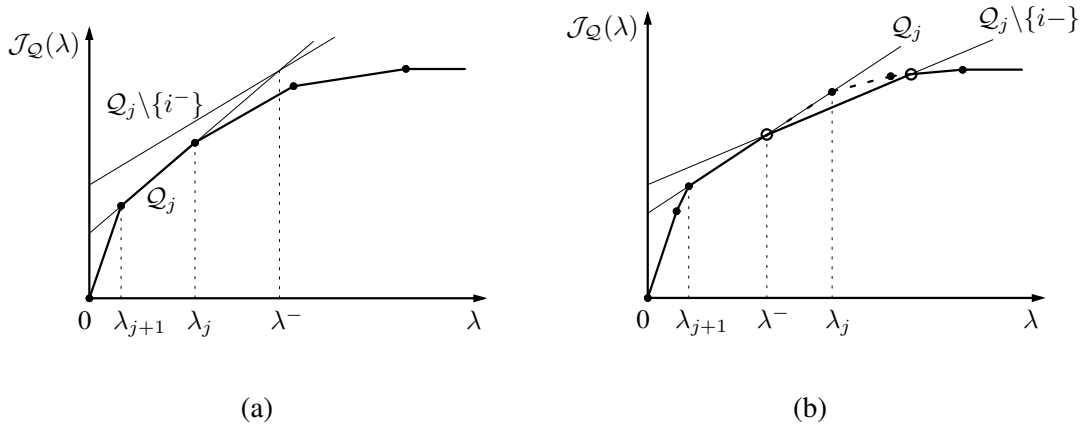


Fig. 6. Best removal from subset \mathcal{Q}_j belonging to the ℓ_0 regularization path. (a) When $\lambda^- \geq \lambda_j$, the new line $\mathcal{Q}_j \setminus \{i^-\}$ lays above the ℓ_0 curve, which is thus not improved. (b) When $\lambda^- < \lambda_j$, $\mathcal{Q}_j \setminus \{i^-\}$ intersects one or two edges of the ℓ_0 curve. $\mathcal{Q}_j \setminus \{i^-\}$ is inserted as a new edge and all edges laying above the line $\mathcal{Q}_j \setminus \{i^-\}$ are removed.

and both lines $\mathcal{J}_{\mathcal{Q}}(\lambda)$ and $\mathcal{J}_{\mathcal{Q} \setminus \{i^-\}}(\lambda)$ intersect at

$$\lambda = \lambda^- \triangleq \mathcal{E}_{\mathcal{Q} \setminus \{i^-\}} - \mathcal{E}_{\mathcal{Q}}. \quad (13)$$

It is easy to check that if $\lambda^- \geq \lambda_j$, the line $\mathcal{J}_{\mathcal{Q} \setminus \{i^-\}}(\lambda)$ lays above the ℓ_0 curve and does not intersect it, thus the ℓ_0 curve is not improved. On the contrary, when $\lambda^- < \lambda_j$, an improvement occurs by inserting $\mathcal{Q} \setminus \{i^-\}$: see Fig. 6.

3) *Refinement of the ℓ_0 regularization path:* When subset \mathcal{Q}_j is explored, either 0, 1, or 2 new supports may be included in the regularization path depending on the values of λ^+ and λ^- . Whenever new supports are included, their explored status is set to 0. The regularization path update simply relies on the computation of line intersections in the plane. When a path refinement occurs, the edges of the previous ℓ_0 curve laying above the new one are erased whether the corresponding supports have been already explored or not: see, *e.g.*, Fig. 5(b). These erased supports are not kept in memory in the further iterations of ℓ_0 -PT.

C. Comments on the ℓ_0 -PT algorithm

The ℓ_0 -PT algorithm is summarized in Tab. III. Similar to CSBR, the stopping conditions involve a maximum cardinality $|\mathcal{Q}_{j-1}| \geq K$ and/or a minimum threshold $\mathcal{E}_{\mathcal{Q}_{j-1}} \leq \varepsilon$ where j denotes the lowest index such that \mathcal{Q}_j is unexplored.

TABLE III

ℓ_0 -PT ALGORITHM. SUBSETS $\mathcal{Q}_k^{\text{PT}}$ ARE DELIVERED IN OUTPUT FOR ANY CARDINALITY k .

Inputs: \mathbf{A} , \mathbf{y} , K and/or ε

$J = 0$, $\mathcal{Q}_0 = \emptyset$, $\lambda_0 = +\infty$, $\lambda_1 = 0$. Set $\text{explored}(0) = 0$ and $j = 0$.

While ($|\mathcal{Q}_j| < K$) and ($\mathcal{E}_{\mathcal{Q}_{j-1}} > \varepsilon$),

Compute i^+ according to (10) and set $\lambda^+ = \mathcal{E}_{\mathcal{Q}_j} - \mathcal{E}_{\mathcal{Q}_j \cup \{i^+\}}$.

If $|\mathcal{Q}_j| > 2$, compute i^- according to (12) and set $\lambda^- = \mathcal{E}_{\mathcal{Q}_j \setminus \{i^-\}} - \mathcal{E}_{\mathcal{Q}_j}$.

Set $\text{explored}(j) = 1$.

If ($\lambda^+ > \lambda_{j+1}$),

Add $\mathcal{Q}_j \cup \{i^+\}$ to the regularization path with status explored set to 0.

Update the regularization path by removing subsets.

End if

If ($\lambda^- < \lambda_j$),

/* When $\mathcal{Q}_j \cup \{i^+\}$ has been included in the ℓ_0 curve: */

If $\mathcal{J}_{\mathcal{Q}_j \setminus \{i^-\}}(\lambda)$ intersects the edges of the ℓ_0 curve,

Add $\mathcal{Q}_j \setminus \{i^-\}$ to the regularization path with status explored set to 0.

Update the regularization path by removing subsets.

End if

End if

Select the lowest j such that $\text{explored}(j) = 0$.

End while

Outputs: • Estimated ℓ_0 regularization path $\{\mathcal{Q}_j, j = 0, \dots, J\}$ and squared errors $\mathcal{E}_{\mathcal{Q}_j}$.

- Critical values $\lambda_j, j = 0, \dots, J$.
- Best explored subsets $\mathcal{Q}_k^{\text{PT}}$ and their squared errors $\mathcal{E}_k^{\text{PT}}$ ($k = 0, \dots, K$).

Let us highlight the main differences between ℓ_0 -PT and CSBR. First, we stress that the current iteration of ℓ_0 -PT is related to an *edge* of the ℓ_0 -curve, *i.e.*, an interval $(\lambda_{j+1}, \lambda_j)$ whereas the current iteration of CSBR is related to a *specific* λ_j -value. Second, we remark that in CSBR, the computation of the next value $\lambda_{j+1} = \lambda^+ \leq \lambda_j$ is only based on the violation of the lower bound of the stopping condition of SBR (8), corresponding to atom insertions. In ℓ_0 -PT, the atom removals ($\lambda^- \geq \lambda_j$) are considered as well. Therefore, the λ -values are not scanned in a decreasing order anymore. This may lead to substantial improvement of the very sparse solutions found in the early iterations within an increased computation time, as we will see hereafter.

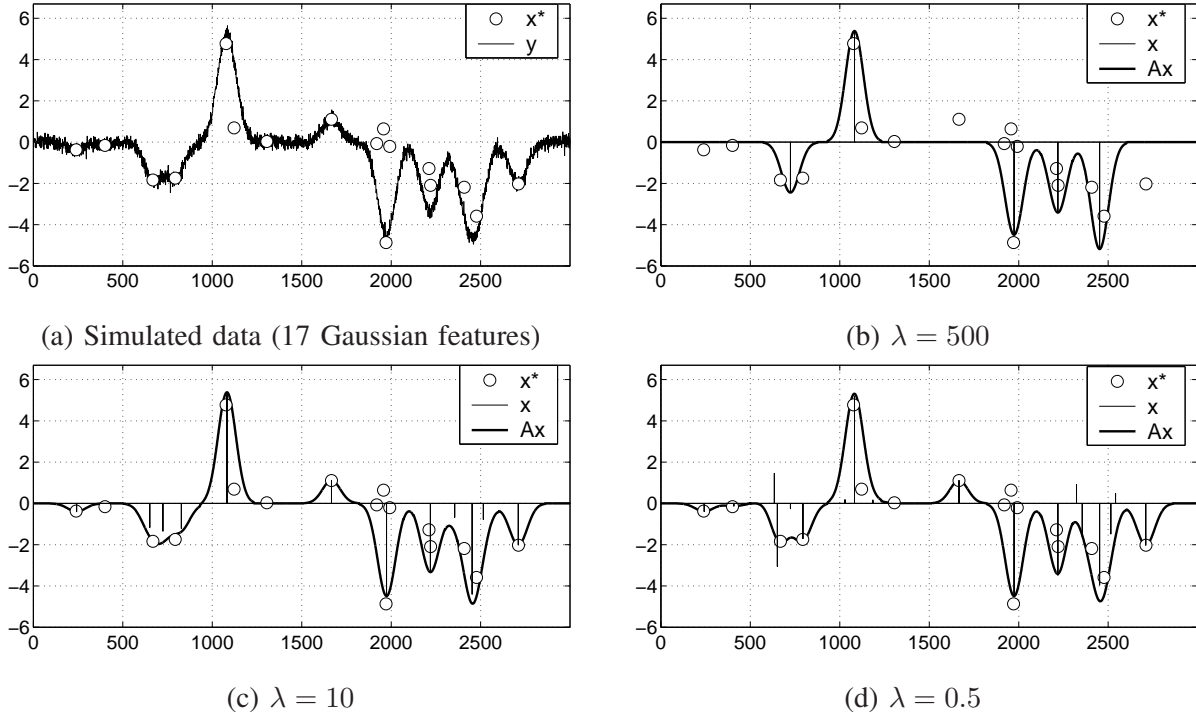


Fig. 7. Sparse deconvolution with a low-pass filter (results excerpted from [3]). (a) Simulated data with 17 Gaussian features. The signal-to-noise ratio is equal to 20 dB. (b,c,d) Spike signals obtained as SBR outputs and related data approximation signals with empirical tuning of λ : $\lambda = 500, 10$ and 0.5 , respectively. The estimated amplitudes \mathbf{x} are represented with vertical bars. Their supports are of size 5, 12, and 18. The computation time remains below 3 seconds in a Matlab implementation specific to deconvolution [3].

V. APPLICATION TO SPARSE SPIKE TRAIN DECONVOLUTION

The proposed algorithms are evaluated on a spike train deconvolution problem of the form $\mathbf{y} = \mathbf{h} * \mathbf{x}^* + \mathbf{b}$ where the impulse response \mathbf{h} is a low-pass filter and the noise \mathbf{b} is assumed to be i.i.d. and Gaussian. The problem rereads $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ where \mathbf{A} is a Toeplitz matrix whose columns are shifted versions of \mathbf{h} . Specifically, \mathbf{h} is a Gaussian filter of standard deviation $\sigma = 50$: the Gaussian pattern induced by a spike $x_i^* \neq 0$ has a width equal to 301 samples. This yields a matrix \mathbf{A} of size 3000×2700 . This sparse signal restoration problem is difficult because the columns of \mathbf{A} are highly correlated, and a number of fast algorithms that are efficient for well-conditioned dictionaries fail in this situation.

Fig. 7 illustrates the behavior of SBR for decreasing λ -values. The simulated data \mathbf{y} are related to an unknown sparse signal \mathbf{x}^* including 17 spikes. Some spike locations are close enough so that the resulting Gaussian features overlap, and other spikes of small amplitudes are drowned in the noise. For the largest

λ , only the main Gaussian features are recovered. While λ decreases, the other features are reconstructed together with some false spike detections. In [3], we advocated that SBR behaves much better than simpler and faster ℓ_0 algorithms such as iterative thresholding algorithms (Iterative Hard Thresholding, CoSaMP, Subspace Pursuit) which have been proposed in the context of compressive sensing, *i.e.*, for relatively well conditioned dictionaries. Unsurprisingly, SBR outperforms OMP based algorithms of lower complexity as well as the basis pursuit denoising algorithms relying on the ℓ_1 relaxation of the ℓ_0 norm. Other authors have drawn similar empirical conclusions regarding the efficiency of OLS based algorithms for ill-conditioned problems [11], [34], [40]. We found that SBR is as efficient as the Iterative Reweighted ℓ_1 algorithm, associated to nonconvex continuous relaxations of the ℓ_0 norm [22]. However, the structure of SBR is simpler (no call to any ℓ_1 subroutine is required) and the number of parameters to tune is much lower: there is a single parameter λ and no arbitrary stopping condition. Although the cost per iteration of SBR is relatively high given the large number of linear inversions per iteration, the number of iterations is very limited (less than 25 iterations for the deconvolution problem of Fig. 7). On the contrary, the cost per iteration of Iterative Hard Thresholding is low but the convergence necessitates at least 10,000 iterations leading to an overall computation time larger than that of SBR.

The following simulations aim to show that (i) CSBR and ℓ_0 -PT improve the SBR efficiency, and (ii) in a practical viewpoint, they may be simpler to use because the empirical tuning of λ (which is the main difficulty when using SBR) is not a limitation anymore, and the use of automatic selection rules is enabled. For simplicity reasons, algorithms are compared in terms of approximation error for the same cardinality. An alternative viewpoint would be to evaluate the supports of the sparse signals in terms of number of good spike detection and false alarms. See *e.g.*, [41] for such comparison of sparse algorithms including SBR. For difficult problems though, these tests may not be informative enough because a very few spikes are exactly recovered by any algorithm. More sophisticated localization tests are non binary and take into account the distance between the true spikes and their wrong estimates [42], [43].

A. Comparison SBR vs CSBR for fixed sparsity level

Let us first illustrate the benefit of CSBR over SBR. Fig. 8(a) compares the supports of cardinality k yielded by OLS and CSBR, and the SBR outputs of same cardinality where SBR has been run for various sparsity levels λ until the cardinality k is found. The results are represented in the plane $(k, \mathcal{E}_{\mathcal{Q}})$. We observe that the CSBR curve lays below the OLS and SBR curves (these three curves are almost identical for $k \leq 16$). Moreover, the SBR curve includes some irregularities indicating a strong sensitivity to small variations of λ around specific λ -values. The cardinality of the SBR output does not systematically

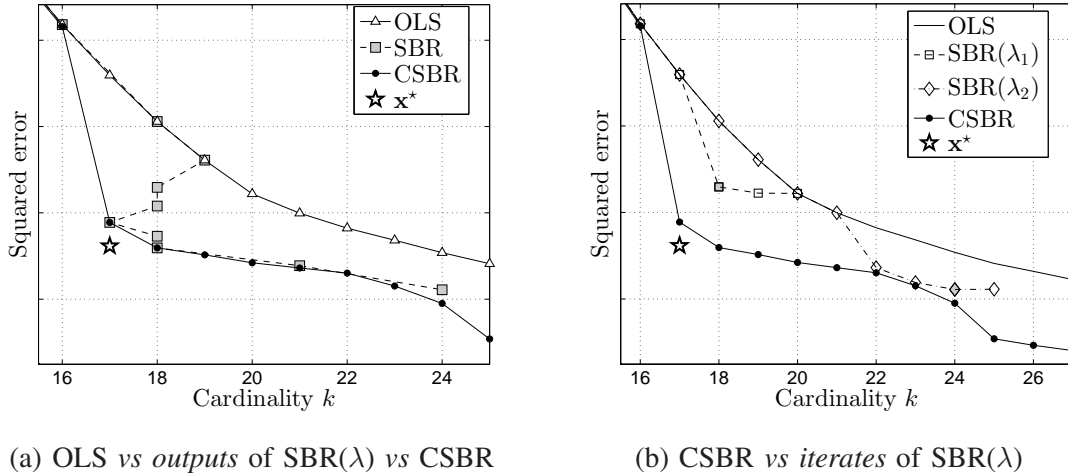


Fig. 8. Comparison of OLS, SBR and CSBR. The results are represented in the plane $(|\mathcal{Q}|, \mathcal{E}_{\mathcal{Q}})$. The star represents the unknown solution: $\|\mathbf{x}^*\|_0 = 17$ and $\|\mathbf{y} - \mathbf{h} * \mathbf{x}^*\|_2^2 = \|\mathbf{b}\|_2^2$. (a) OLS and CSBR are run once with the stopping conditions $K = 25$ and $\varepsilon = 0$. SBR is run repeatedly for all values λ_j obtained in output of CSBR (estimated regularization path). The supports \mathcal{Q}_k ($k \leq K$) yielded by OLS and CSBR are compared with the outputs $\mathcal{Q}_j = \text{SBR}(\emptyset; \lambda_j)$. (b) Same execution of OLS and CSBR. SBR is run for two values $\lambda_1 > \lambda_2$. For each execution, the SBR iterates are all represented (squares/diamonds). The output supports $\text{SBR}(\emptyset; \lambda_1)$ and $\text{SBR}(\emptyset; \lambda_2)$ are of cardinalities 18 and 24, respectively (in grey color).

increase while λ decreases: it is successively equal to 19, 18, 17, 18, and 21 (Fig. 8(a)).

The obvious advantage of CSBR over SBR is that a single execution of CSBR delivers solutions $(\mathcal{Q}_k^{\text{CSBR}}, \mathcal{E}_k^{\text{CSBR}})$ for any k without empirical tuning of parameters (except for the usual stopping criteria K and/or ε). On the contrary, the SBR output support is related to a single λ , whose tuning may be tricky. Although SBR works for fixed λ , one could think of SBR as a continuation method in the cardinality domain because consecutive supports are nested. In other words, storing the SBR iterates obtained while running $\text{SBR}(\emptyset; \lambda)$ yields subsets $\mathcal{Q}_k^{\text{SBR}}$ for consecutive values of k . We found that this strategy is ineffective: the best iterates provided by SBR have an approximation error substantially larger than the CSBR solutions of same cardinality. See Fig. 8(b) where SBR has been run for two sparsity levels $\lambda_1 > \lambda_2$.

B. Comparison of continuation algorithms for variable sparsity levels

We compare four strategies to reconstruct supports for all cardinalities k : OLS, CSBR and ℓ_0 -PT for ℓ_2 - ℓ_0 continuation and the homotopy algorithm solving the ℓ_2 - ℓ_1 continuation problem. The ℓ_1 regularization not only induces a sparsity constraint but also a penalty on the amplitudes $|x_i|$. Therefore, the squared errors $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ related to the homotopy solutions do not match the best possible approximation $\mathcal{E}_{\mathcal{Q}}$,

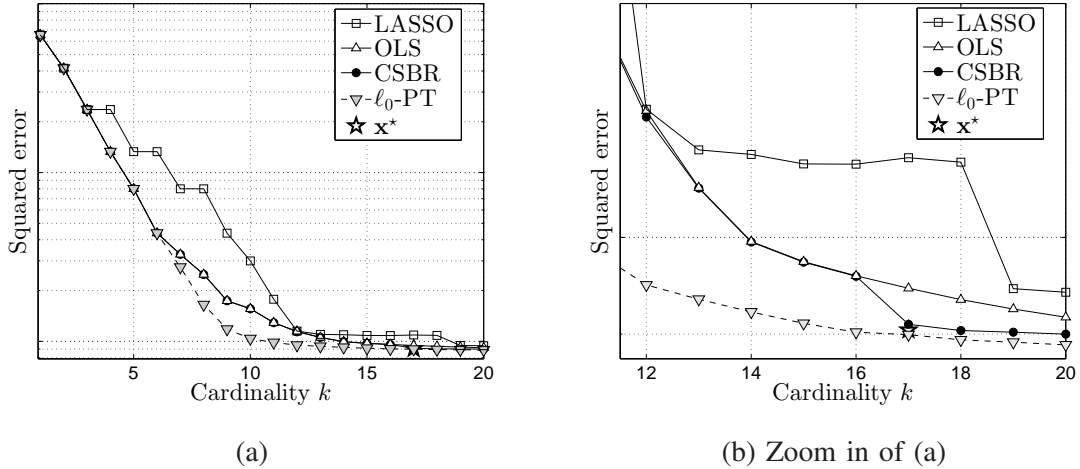


Fig. 9. Comparison of OLS, CSBR, and ℓ_0 -PT for ℓ_2 - ℓ_0 continuation, and the homotopy algorithm for ℓ_2 - ℓ_1 continuation (LASSO).

where \mathcal{Q} is the support of \mathbf{x} . To make the comparison fair with the orthogonal greedy algorithms based on the ℓ_0 -norm, a debiasing post-processing is necessary. The evaluation of $\mathcal{E}_{\mathcal{Q}}$ requires to compute an orthogonal projection of the data \mathbf{y} for any support \mathcal{Q} obtained by ℓ_1 homotopy. Fig. 9(a) illustrates that even with debiasing, the homotopy solutions are less accurate than those of OLS, CSBR and ℓ_0 -PT. ℓ_0 -PT substantially improves the OLS and CSBR performance but the computation cost is increased. With the stopping parameters $K = 34$ and $\varepsilon = 0$, 34 iterations of OLS are proceeded versus 58 (number of single replacements from the initial empty support) for CSBR and 113 for ℓ_0 -PT. Regarding ℓ_0 -PT, 8% of the iterations are ineffective: no single replacement improves the current regularization path. For 62% of the iterations, a new support is generated and included in the current regularization path. Finally, two new supports are included for 30% of the iterations. Considering that the cost per iteration of ℓ_0 -PT is almost identical to that of CSBR (except for the ℓ_0 -curve update), the price to pay for the better performance with ℓ_0 -PT is roughly a double computation cost.

Fig. 10 provides an insight on the behavior of ℓ_0 -PT and CSBR. During the first 25 iterations, ℓ_0 -PT mainly operates atom selections similar to OLS. The explored subsets are thus of increasing cardinality and the sparsity level λ is decreasing (Figs. 10(c,d)). From iterations 25 to 40, the very sparse solutions previously obtained ($k = 20, 19, \dots, 7$) are improved as the algorithm performs a series of atom de-selections. They are being improved again around iteration 80. On the contrary, CSBR explores supports of “globally” increasing cardinalities (although some de-selections are done). The sparsest solutions are never improved because CSBR works for decreasing λ -values (Figs. 10(a,b)).

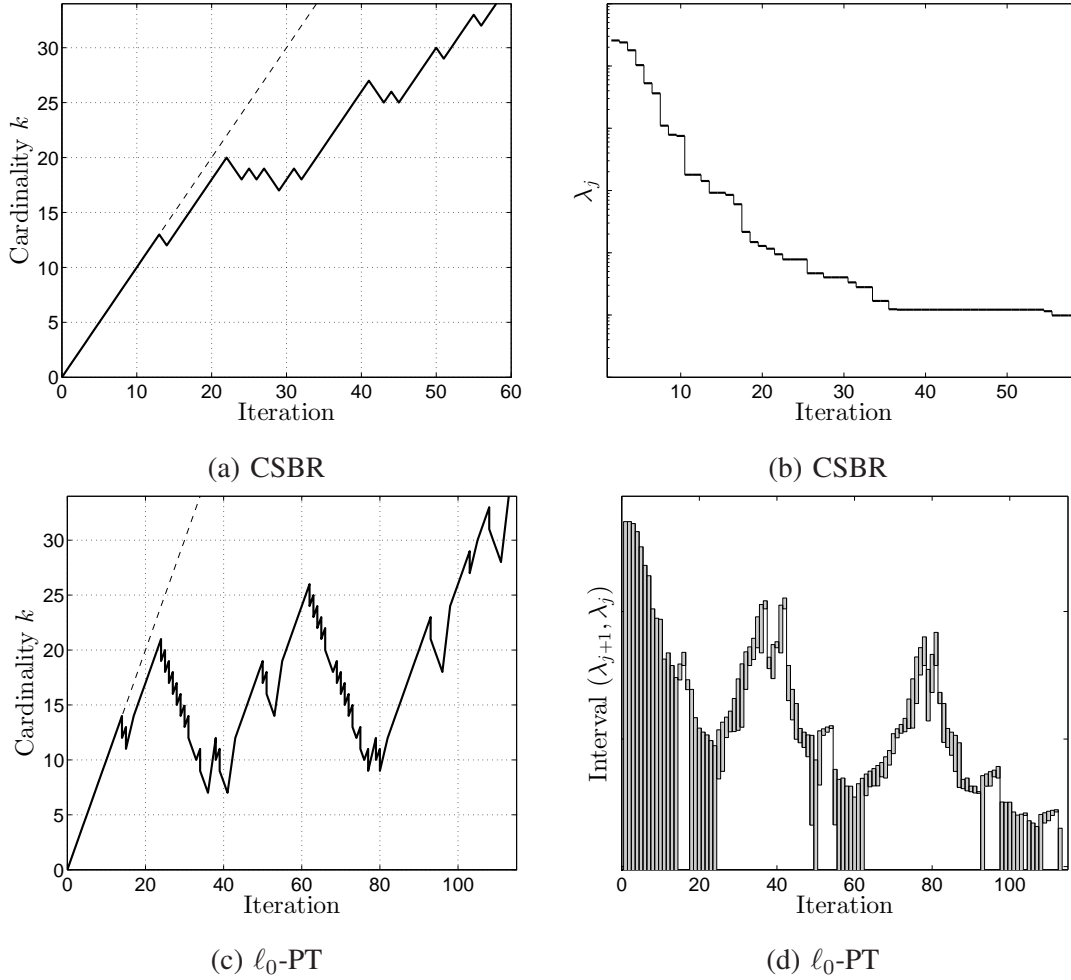


Fig. 10. Empirical behavior of CSBR and ℓ_0 -PT. (a) CSBR: cardinality of the current support after each single replacement (iteration number in horizontal axis) during the calls to SBR. (b) CSBR: critical values λ_j explored, represented in log-scale. SBR is executed for each λ_j , and the number of single replacements for fixed λ_j matches the length of the horizontal steps in the figure. (c) ℓ_0 -PT: cardinality of the supports appended to the regularization path during the ℓ_0 -PT iterations. At each iteration, 0, 1 or 2 supports are included. Vertical steps appear whenever two supports are simultaneously included. (d) ℓ_0 -PT: representation in log-scale of the sparsity interval $(\lambda_{j+1}, \lambda_j)$ scanned during the current iteration (grey color). When the grey bars reach the bottom of the image, the lower bound equals $\lambda_{j+1} = 0$.

C. Model order selection

The proposed continuation algorithms are naturally compatible with most classical methods of model order selection [44], [45] because they provide a single (sub-optimal) candidate solution Q_k^{CSBR} and Q_k^{PT} for each cardinality $k = 0, \dots, K$. Here, we assume that the variance of the observation noise is unknown, and we consider two categories of cost functions for the estimation of k . The first take the

form $\arg \min_k \{m \log(\mathcal{E}_k) + \alpha k\}$, where m is the size of \mathbf{y} and α equals 2, $\log m$, and $2 \log \log m$ for the Akaike (AIC), Minimum Description Length (MDL) and Hannan and Quinn criteria, respectively [44]⁴. The second category are cross-validation criteria [46], [47]. In the leave-one-out version, they read $\arg \min_k \|\mathbf{y} - \hat{\mathbf{y}}(k)\|_2^2/m$ where the i -th entry of $\hat{\mathbf{y}}(k)$ is defined as the i -th entry of $\mathbf{A}\mathbf{x}(\mathbf{y}^{[i]}, k)$ with $\mathbf{y}^{[i]}$ the reduced observation signal obtained by removing the single observation y_i from \mathbf{y} .

The sparse approximation framework allows one to derive simplified expressions of the cross-validation criterion and its generalized versions. The interested reader is referred to the books [9, Chap. 5] and [45] for more details. For sparse deconvolution problems, we found that the Akaike and cross validation criteria severely over-estimate the expected number of spikes (45 and 50 spikes are detected with CSBR and ℓ_0 -PT, respectively whereas the unknown signal \mathbf{x}^* only includes 17 spikes). Their generalized versions behave similarly. The MDL criterion yields the most realistic results (25 and 20 spikes are found with CSBR and ℓ_0 -PT, respectively). Furthermore, the number of spikes is underestimated for higher noise levels: 8 spikes are found with both CSBR and ℓ_0 -PT for a signal-to-noise ratio of 0 dB. This behavior is relevant because for highly noisy data, the smallest spikes are drowned in the noise. Thus, one cannot expect to detect them.

VI. CONCLUSION

The choice of a relevant sparse approximation algorithm relies on a trade-off between the desired performance and the computation time one is ready to spend. The proposed bidirectional OLS-based algorithms are relatively expensive but very well suited to inverse problems inducing highly correlated dictionaries. A reason is that they have the capacity to “escape” from local minimizers of the cost function $\mathcal{J}(\mathbf{x}; \lambda) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0$ for a given sparsity level λ [3]. Actually, each iterate of the SBR algorithm is a local minimizer of \mathcal{J} . This behavior is in contrast with other classical sparse algorithms which cannot escape from a local minimizer of \mathcal{J} . The efficiency of SBR is acknowledged by other researchers [27]. Here, we have derived two new algorithms from SBR, namely CSBR and ℓ_0 -PT, and we have shown that their efficiency is significantly increased over SBR. Moreover, the proposed algorithms provide solutions for a continuum of λ -values and contiguous cardinalities, enabling the utilization of any classical order selection method based on the optimization of a simple criterion depending on the order k . We found that the MDL criterion specifically yields accurate estimates of the cardinality $\|\mathbf{x}\|_0$ in contrast to the

⁴Note that when the noise variance is known, the first term $\log(\mathcal{E}_k)$ appearing in the cost function is replaced by a quadratic term proportional to \mathcal{E}_k , leading to ℓ_0 penalized least-square formulations.

other simple criteria we tested. Other more elaborate criteria proposed recently could be considered as well [48].

Our perspectives include the proposal of bidirectional search algorithms for mixed ℓ_2 - ℓ_0 optimization that will be faster than SBR and potentially more efficient for specific inverse problems, *e.g.*, sparse deconvolution. In the standard version of SBR, CSBR and ℓ_0 -PT presented here, a single replacement refers to the insertion or removal of a dictionary element. The cost of an iteration is essentially related to the n linear system resolutions done to test the single replacements for all dictionary atoms. The proposed algorithms obviously remain valid when working with a larger neighborhood, *e.g.*, when testing the replacement of two atoms simultaneously, but their complexity becomes huge. To avoid such numerical explosion, one may rather choose not to carry out all replacement tests, but only some tests that are likely to be effective. Monodirectional extensions of OMP and OLS were recently proposed in this spirit [34] and deserve consideration for proposing efficient bidirectional algorithms.

APPENDIX A

PROPERTIES OF MIXED ℓ_2 - ℓ_0 REGULARIZATION PATHS

In this appendix, we prove that the optimal ℓ_2 - ℓ_0 regularization path \mathcal{X}_p (see Definition 1) is piecewise constant (Theorem 1) and is a subset of the ℓ_0 constrained regularization path \mathcal{X}_c (Theorem 2).

A. Proof of Theorem 1

Let $\lambda \mapsto \mathcal{J}(\lambda)$ refer to the ℓ_0 curve. For finite λ , $\mathcal{J}(\lambda)$ is the minimum of $\mathcal{J}_{\mathcal{Q}}(\lambda)$ over \mathcal{Q} . Function $\lambda \mapsto \mathcal{J}(\lambda)$ is continuous, increasing and piecewise affine as the minimum of a finite set of increasing and affine functions $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda)$. The critical values λ_i^* , introduced in Theorem 1 and Fig. 1, delimit the intervals on which \mathcal{J} is affine. In particular, for $\lambda \leq \lambda_1^*$, $\mathcal{X}_p(\lambda)$ gathers the supports of the sparsest unconstrained least-square minimizers while for $\lambda \geq \lambda_1^*$, $\mathcal{X}_p(\lambda)$ reduces to the empty support and $\mathcal{J}(\lambda) = \|\mathbf{y}\|_2^2$.

Let us now prove Theorem 1. Simultaneously, we will prove the additional technical result.

Lemma 1 *If $\lambda_{i+1}^* > 0$, $\mathcal{X}_p(\lambda) \subset \mathcal{X}_p(\lambda_{i+1}^*) \cap \mathcal{X}_p(\lambda_i^*)$ for $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$, and when $\lambda \in (0, \lambda_1^*)$, $\mathcal{X}_p(\lambda) \subset \mathcal{X}_p(\lambda_1^*)$.*

Proof of Theorem 1 and Lemma 1: Function $\mathcal{J}(\lambda)$ is affine on a given interval $[\lambda_{i+1}^*, \lambda_i^*]$ and reads $\mathcal{J}(\lambda) = \mathcal{J}_{\mathcal{Q}_i}(\lambda) = \mathcal{E}_{\mathcal{Q}_i} + \lambda|\mathcal{Q}_i|$ where \mathcal{Q}_i is a subset of $\{1, \dots, n\}$. Let us show that for $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$, $\mathcal{X}_p(\lambda)$ is a constant set.

Let $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$ and $\mathcal{Q} \in \mathcal{X}_p(\lambda)$. Then, $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{J}_{\mathcal{Q}_i}(\lambda)$. Both lines $\mathcal{J}_{\mathcal{Q}}$ and $\mathcal{J}_{\mathcal{Q}_i}$ necessarily coincide, otherwise they would intersect at λ , and $\mathcal{J}_{\mathcal{Q}}$ would lay below $\mathcal{J}_{\mathcal{Q}_i}$ on either $(\lambda_{i+1}^*, \lambda)$ or (λ, λ_i^*) which contradicts the definition of \mathcal{Q}_i .

We have shown that $\mathcal{Q} \in \mathcal{X}_p(\lambda)$ implies that $\mathcal{J}_{\mathcal{Q}}(\lambda') = \mathcal{J}(\lambda')$, and hence $\mathcal{Q} \in \mathcal{X}_p(\lambda')$ for all $\lambda' \in [\lambda_{i+1}^*, \lambda_i^*]$. Thus, the content of $\mathcal{X}_p(\lambda)$ does not depend on λ when $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$. Moreover, we have shown that $\mathcal{X}_p(\lambda) \subset \mathcal{X}_p(\lambda_{i+1}^*) \cap \mathcal{X}_p(\lambda_i^*)$ since $\mathcal{J}_{\mathcal{Q}}(\lambda') = \mathcal{J}(\lambda')$ holds for $\lambda' = \lambda_{i+1}^*$ and λ_i^* . ■

B. Proof of Theorem 2

Proof: The first result is obvious: for any λ and for $\mathcal{Q} \in \mathcal{X}_p(\lambda)$, we have $\mathcal{Q} \in \mathcal{X}_c(|\mathcal{Q}|)$. Otherwise, there would exist \mathcal{Q}' with $|\mathcal{Q}'| \leq |\mathcal{Q}|$ and $\mathcal{E}_{\mathcal{Q}'} < \mathcal{E}_{\mathcal{Q}}$. Then, $\mathcal{J}_{\mathcal{Q}'}(\lambda) < \mathcal{J}_{\mathcal{Q}}(\lambda)$ would contradict $\mathcal{Q} \in \mathcal{X}_p(\lambda)$. Let us show that for any i , $\exists k_i : \forall \lambda \in (\lambda_{i+1}^*, \lambda_i^*)$, $\mathcal{X}_p(\lambda) \subset \mathcal{X}_c(k_i)$.

Let $\mathcal{Q} \in \mathcal{X}_p(\lambda)$ for some $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$. Theorem 1 implies that $\mathcal{Q} \in \mathcal{X}_p(\lambda)$ for any $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$. Therefore, $\mathcal{J}(\lambda) = \mathcal{J}_{\mathcal{Q}}(\lambda)$ for all $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$ and the slope of $\mathcal{J}_{\mathcal{Q}}$, *i.e.*, $|\mathcal{Q}|$, is constant whatever $\mathcal{Q} \in \mathcal{X}_p(\lambda)$ and $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$. Let us denote this constant by $k_i = |\mathcal{Q}|$. According to the preceding paragraph, $\mathcal{Q} \in \mathcal{X}_p(\lambda)$ implies that $\mathcal{Q} \in \mathcal{X}_c(|\mathcal{Q}|) = \mathcal{X}_c(k_i)$.

Let us prove the reverse inclusion. Let $\lambda \in (\lambda_{i+1}^*, \lambda_i^*)$ and $\mathcal{Q} \in \mathcal{X}_c(k_i)$. First, we have $|\mathcal{Q}| \leq k_i$. Second, for any $\mathcal{Q}' \in \mathcal{X}_p(\lambda)$, we have $\mathcal{E}_{\mathcal{Q}} = \mathcal{E}_{\mathcal{Q}'}$ because $\mathcal{X}_p(\lambda) \subset \mathcal{X}_c(k_i)$. Finally, $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda|\mathcal{Q}| \leq \mathcal{E}_{\mathcal{Q}'} + \lambda k_i = \mathcal{J}_{\mathcal{Q}'}(\lambda)$. $\mathcal{Q}' \in \mathcal{X}_p(\lambda)$ implies that $\mathcal{Q} \in \mathcal{X}_p(\lambda)$. ■

REFERENCES

- [1] B. K. Natarajan, “Sparse approximate solutions to linear systems”, *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.
- [2] M. Nikolova, “Description of the minimizers of least squares regularized with ℓ_0 norm. Uniqueness of the global minimizer”, *SIAM J. Imaging Sci.*, vol. 6, no. 2, pp. 904–937, May 2013.
- [3] C. Soussen, J. Idier, D. Brie, and J. Duan, “From Bernoulli-Gaussian deconvolution to sparse signal restoration”, *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4572–4584, Oct. 2011.
- [4] C. Herzet and A. Drémeau, “Bayesian pursuit algorithms”, Research Report, INRIA Rennes Bretagne Atlantique - Télécom ParisTech, Rennes, France, Jan. 2014.
- [5] J. A. Tropp and S. J. Wright, “Computational methods for sparse solution of linear inverse problems”, *Proc. IEEE, invited paper (Special Issue “Applications of sparse representation and compressive sensing”)*, vol. 98, no. 5, pp. 948–958, June 2010.
- [6] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries”, *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [7] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition”, in *Proc. 27th Asilomar Conf. on Signals, Systems and Computers*, Nov. 1993, vol. 1, pp. 40–44.

- [8] S. Chen, S. A. Billings, and W. Luo, “Orthogonal least squares methods and their application to non-linear system identification”, *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.
- [9] A. J. Miller, *Subset selection in regression*, Chapman and Hall, London, UK, 2nd edition, Apr. 2002.
- [10] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, “Forward sequential algorithms for best basis selection”, *IEE Proc. Vision, Image and Signal Processing*, vol. 146, no. 5, pp. 235–244, Oct. 1999.
- [11] L. Rebollo-Neira and D. Lowe, “Optimized orthogonal matching pursuit approach”, *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 137–140, Apr. 2002.
- [12] T. Blumensath and M. E. Davies, “Iterative thresholding for sparse approximations”, *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 629–654, Dec. 2008.
- [13] T. Blumensath, “Accelerated iterative hard thresholding”, *Signal Process.*, vol. 92, pp. 752–756, 2012.
- [14] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction”, *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.
- [15] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”, *Appl. Comp. Harmonic Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.
- [16] M. A. Davenport, D. Needell, and M. B. Wakin, “Signal space CoSaMP for sparse recovery with redundant dictionaries”, *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6820–6829, Oct. 2013.
- [17] M. A. Efron, “Multiple regression analysis”, in *Mathematical Methods for Digital Computers*, A. Ralston and H. S. Wilf, Eds., vol. 1, pp. 191–203. Wiley, New York, 1960.
- [18] K. N. Berk, “Forward and backward stepping in variable selection”, *J. Statist. Comput. Simul.*, vol. 10, no. 3-4, pp. 177–185, Apr. 1980.
- [19] T. Zhang, “Adaptive forward-backward greedy algorithm for learning sparse representations”, *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, July 2011.
- [20] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems”, *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
- [21] M. Zibulevsky and M. Elad, “ $\ell_1 - \ell_2$ optimization in signal and image processing”, *IEEE Trans. Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.
- [22] E. J. Candès, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization”, *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.
- [23] N. Mourad and J. P. Reilly, “Minimizing nonconvex functions for sparse vector reconstruction”, *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3485–3496, July 2010.
- [24] D. P. Wipf and S. Nagarajan, “Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions”, *IEEE J. Sel. Top. Signal Process. (Special issue on Compressive Sensing)*, vol. 4, no. 2, pp. 317–329, Apr. 2010.
- [25] A. Gholami and S. M. Hosseini, “A general framework for sparsity-based denoising and inversion”, *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5202–5211, Nov. 2011.
- [26] I. Ramírez and G. Sapiro, “Universal regularizers for robust sparse coding and modeling”, *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3850–3864, Sept. 2012.
- [27] I. Selesnick and I. Bayram, “Sparse signal estimation by maximally sparse convex optimization”, *to appear in IEEE Trans. Signal Process.*, pp. 1–31, Jan. 2014.
- [28] D. L. Donoho and Y. Tsaig, “Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse”, *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

- [29] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression”, *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.
- [30] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering”, *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr. 2004.
- [31] I. Das and J. E. Dennis, “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems”, *Structural optimization*, vol. 14, no. 1, pp. 63–69, Aug. 2007.
- [32] P. M. T. Broersen, “Subset regression with stepwise directed search”, *J. R. Statist. Soc. C*, vol. 35, no. 2, pp. 168–177, 1986.
- [33] D. Haugland, “A bidirectional greedy heuristic for the subspace selection problem”, in *Engineering stochastic local search algorithms. Designing, implementing and analyzing effective heuristics*, T. Stützle, M. Birattari, and H. H. Hoos, Eds., Berlin, Germany, Sept. 2007, vol. 4638 of *Lect. Notes Comput. Sci.*, pp. 162–176, Springer Verlag.
- [34] S. Chatterjee, D. Sundman, M. Vehkaperä, and M. Skoglund, “Projection-based and look-ahead strategies for atom selection”, *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 634–647, Feb. 2012.
- [35] J. Duan, C. Soussen, D. Brie, and J. Idier, “A continuation approach to estimate a solution path of mixed L2-L0 minimization problems”, in *Signal Processing with Adaptive Sparse Structured Representations (SPARS workshop)*, Saint-Malo, France, Apr. 2009, pp. 1–6.
- [36] E. Wasserstrom, “Numerical solutions by the continuation method”, *SIAM Rev.*, vol. 15, no. 1, pp. 89–119, Jan. 1973.
- [37] J. Trzasko and A. Manduca, “Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0 -minimization”, *IEEE Trans. Medical Imaging*, vol. 8, no. 1, pp. 106–121, Jan. 2009.
- [38] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, “A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm”, *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.
- [39] D. M. Malioutov, M. Cetin, and A. S. Willsky, “Homotopy continuation for sparse signal representation”, in *Proc. IEEE ICASSP*, Philadelphia, PA, Mar. 2005, vol. V, pp. 733–736.
- [40] P. Dymarski, N. Moreau, and G. Richard, “Greedy sparse decompositions: a comparative study”, *Eurasip Journal on Advances in Signal Processing*, vol. 2011, no. 34, pp. 1–16, Aug. 2011.
- [41] S. Bourguignon, C. Soussen, H. Carfantan, and J. Idier, “Sparse deconvolution: Comparison of statistical and deterministic approaches”, in *IEEE Workshop Stat. Sig. Proc.*, Nice, France, June 2011, pp. 317–320.
- [42] M. C. van Rossum, “A novel spike distance”, *Neural Computation*, vol. 13, no. 4, pp. 751–763, Apr. 2001.
- [43] E. Carcreff, S. Bourguignon, J. Idier, and L. Simon, “Resolution enhancement of ultrasonic signals by up-sampled sparse deconvolution”, in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 6511–6515.
- [44] P. Stoica and Y. Selén, “Model-order selection: a review of information criterion rules”, *IEEE Trans. Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, July 2004.
- [45] Y. Wang, “Model selection”, in *Handbook of Computational Statistics*, J. E. Gentle, W. Härdle, and Y. Mori, Eds., Berlin, Aug. 2004, vol. 1, pp. 437–466, Springer-Verlag.
- [46] G. Wahba, “Practical approximate solutions to linear operator equations when the data are noisy”, *SIAM J. Num. Anal.*, vol. 14, no. 4, pp. 651–667, 1977.
- [47] G. H. Golub, M. Heath, and G. Wahba, “Generalized cross-validation as a method for choosing a good ridge parameter”, *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979.
- [48] P. Stoica and P. Babu, “Model order estimation via penalizing adaptively the likelihood (PAL)”, *Signal Process.*, vol. 93, no. 11, pp. 2865–2871, Nov. 2013.