A Continuation Approach to Estimate a Solution Path of Mixed L2-L0 Minimization Problems

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

*Centre de Recherche en Automatique de Nancy
Nancy-University, CNRS. Boulevard des Aiguillettes,
B.P. 70239, 54506 Vandœuvre-lès-Nancy, France.
E-mail: `FirstName.LastName@cran.uhp-nancy.fr`.

†Institut de Recherche en Communication et Cybernétique
de Nantes (UMR CNRS 6597). 1 rue de la Noë,
B.P. 92101, 44321 Nantes Cedex 3, France.
E-mail: `Jerome.Idier@irccyn.ec-nantes.fr`.

***Abstract*—The approximation of a signal using a limited number of dictionary elements is stated as an L0-constrained or an L0-penalized least-square problem. We first give the working assumptions and then propose the heuristic Single Best Replacement (SBR) algorithm for the penalized problem. It is inspired by the Single Most Likely Replacement (SMLR) algorithm, initially proposed in the context of Bernoulli-Gaussian deconvolution. Then, we extend the SBR algorithm to a continuation version estimating a whole solution path, *i.e.*, a series of solutions depending on the level of sparsity. The continuation algorithm, up to a slight adaptation, also provides an estimate of a solution path of the L0-constrained problem. The effectiveness of this approach is illustrated on a sparse signal deconvolution problem.**

I. Introduction

The goal of sparse approximation is to represent an observation by a linear combination of a limited number of given signals. The signals are chosen from a set containing elementary signals, often referred to as a dictionary. Typical dictionaries include the Fourier dictionary, the wavelet dictionary, the Gabor dictionary [1]. Sparse approximation can be formulated as an L0-constrained least-square problem of the form $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ subject to $\|\boldsymbol{x}\|_0 \leqslant k$, where $\boldsymbol{A}$ stands for the given dictionary and $\|\boldsymbol{x}\|_0$ is the number of nonzero entries of $\boldsymbol{x}$, often referred to as the L0-norm of $\boldsymbol{x}$. This problem will be called *constrained L2-L0*. The main difficulty when minimizing the least-square error with the L0-norm constraint is to find the support of $\boldsymbol{x}$. Indeed, once the support is known, the amplitude estimation reduces to a simple unconstrained least-square optimization problem.

Searching for the support of $\boldsymbol{x}$ results in a discrete problem which, except for very specific cases, is known to be NP-hard [2]. Therefore, the optimal solution is generally unavailable unless an exhaustive search is performed. There are two main heuristic strategies to solve this problem sub-optimally.

The first strategy is based on a continuous approximation of the L0-norm, leading to a continuous minimization problem. Among the convex approximations, the L1-norm [3] has received considerable attention because optimal and fast algorithms are available to compute the solution path of the L2-L1 problem [4, 5]. Moreover, some conditions [6, 7] were found under which these solutions share the same supports as those of the L2-L0 problem. However, if these conditions are not fulfilled, there is no guarantee that the exact solutions of the approximate problem are close to those of the constrained L2-L0 problem.

The second strategy directly considers the constrained L2-L0 problem, but only a limited number of the candidate supports are explored. The most classical algorithms (Matching Pursuit [1], Orthogonal Matching Pursuit [8] and Orthogonal Least Squares [9, 10]) share a common feature: the support is gradually increased by one element to ensure the least-square cost decrease. This feature is also their limitation as a false detection of a support element can never be compensated by its further deletion.

In this paper, we develop an algorithm which allows the addition or deletion of an element of the current support. Our approach is based on the L0-penalized least-square formulation (in brief, the penalized L2-L0 formulation) $\min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x};\lambda) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0$. Contrarily to the algorithms dedicated to the constrained L2-L0 problem, for $\lambda > 0$, a decrease of $\mathcal{J}(\boldsymbol{x};\lambda)$ is possible when deleting one element from the support of $\boldsymbol{x}$. This happens when the decrease of the penalization term, *i.e.*, λ, is larger than the increase of the least-square term.

The penalized L2-L0 problem is closely related to maximum *a posteriori* estimation using a Bernoulli-Gaussian model as prior for a sparse signal [11]. Indeed, the Bayesian formulation of the Bernoulli-Gaussian signal restoration yields a penalized criterion involving the L0-norm of the sparse signal. To solve this problem, the deterministic Single Most Likely Replacement (SMLR) algorithm has proved to be effective [12, 13]. SMLR is an iterative algorithm increasing or reducing the current support by one element at a time. Inspired by this algorithm, we introduce our Single Best Replacement (SBR) algorithm dedicated to the penalized L2-L0 problem (Section III). Then, we extend the SBR algorithm in a continuation version estimating a whole solution path, *i.e.*, for any level of sparsity λ (Section IV). The continuation algorithm, up to a slight adaptation, also provides an estimated solution path for the constrained L2-L0 problem, which may differ from that of the penalized problem, since the L0-norm is not a convex function. Finally, simulation results on a sparse signal deconvolution problem are shown and analyzed in Section V.

II. Problem statement

A. Subset selection

Given an observation vector $\boldsymbol{y} \in \mathbb{R}^m$, we want to select a few columns $\boldsymbol{a}_i$ from a dictionary $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$ to approximate $\boldsymbol{y}$ by $\hat{\boldsymbol{y}} = \boldsymbol{A}\hat{\boldsymbol{x}}$, where $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ describes the selection and the weights of the columns $\boldsymbol{a}_i$. The quality of approximation relies on a tradeoff between the approximation residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ and the level of sparsity of $\boldsymbol{x}$, *i.e.,* the L0-norm of $\boldsymbol{x}$.

B. L2-L0 optimization

Sparse signal approximation can be expressed in terms of minimization problems:

— the constrained L2-L0 problem whose goal is to approximate $\boldsymbol{y}$ at best by using no more than k columns:

$$\mathcal{X}_c(k) = \arg\min_{\|\boldsymbol{x}\|_0 \leqslant k} \{\mathcal{E}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2\}; \qquad \text{(L2L0C)}$$

— the penalized L2-L0 problem:

$$\mathcal{X}_p(\lambda) = \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} \{\mathcal{J}(\boldsymbol{x}, \lambda) = \mathcal{E}(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_0\}. \qquad \text{(L2L0P)}$$

Here, λ is a hyperparameter controlling the tradeoff between the approximation residual and the level of sparsity.

Both problems are addressed whatever the size of $\boldsymbol{A}$: m can be either smaller or larger than n. The notations $\mathcal{X}_c(k)$ and $\mathcal{X}_p(\lambda)$ designate the sets of minimizers of (L2L0C) and (L2L0P). Both are subsets of $\mathbb{R}^n$ and not necessarily singletons. Moreover, it can be shown [14] that the dependence of the set $\mathcal{X}_p(\lambda)$ with respect to λ ($\lambda \geqslant 0$) is piecewise constant, with a finite number of intervals. Denoting by $\lambda_i^\star$ ($i = 1, \ldots, I$) the critical λ-values at which the content of $\mathcal{X}_p(\lambda)$ is changing, sorted by increasing order with $\lambda_1^\star = 0$ and $\lambda_I^\star = +\infty$, the sets $\mathcal{X}_p(\lambda)$ are constant on each open interval $(\lambda_i^\star, \lambda_{i+1}^\star)$.

Remark 1. *The solution paths* $\cup_{k \in \mathbb{N}} \mathcal{X}_c(k)$ *and* $\cup_{\lambda \geqslant 0} \mathcal{X}_p(\lambda)$ *generally do not coincide, although the penalized solution path is included in the constrained solution path. The nonequivalence between both is a consequence of the non-convexity of the L0-norm [15]. For* $\lambda \neq \{\lambda_1^\star, \ldots, \lambda_I^\star\}$, *it can be shown that there exists* k *such that* $\mathcal{X}_p(\lambda) = \mathcal{X}_c(k)$. *For more details and technical proofs, we refer the reader to [14].*

C. Notion of active set and related optimization problems

Given a subset $\mathcal{A}$ of $\{1, \ldots, n\}$, we define two minimization problems which are sub-problems of (L2L0C) and (L2L0P):

— the minimization of $\mathcal{E}(\boldsymbol{x})$ over the support $\mathcal{A}$:

$$\mathcal{E}_{\mathcal{A}} = \min_{\{\boldsymbol{x} | \mathcal{A}(\boldsymbol{x}) \subseteq \mathcal{A}\}} \mathcal{E}(\boldsymbol{x}); \tag{1}$$

— for a given λ-value ($\lambda \geqslant 0$), the minimization of $\mathcal{J}(\boldsymbol{x}; \lambda)$ over the support $\mathcal{A}$:

$$\min_{\{\boldsymbol{x} | \mathcal{A}(\boldsymbol{x}) \subseteq \mathcal{A}\}} \mathcal{J}(\boldsymbol{x}; \lambda). \tag{2}$$

The notation $\mathcal{A}(\boldsymbol{x})$ denotes the support of $\boldsymbol{x}$, therefore, the constraint $\{\boldsymbol{x} | \mathcal{A}(\boldsymbol{x}) \subseteq \mathcal{A}\}$ indicates that only the columns $\boldsymbol{a}_i$ of $\boldsymbol{A}$ such that $i \in \mathcal{A}$ are taken into account. In the following, these columns $\boldsymbol{a}_i$ will be referred to as the **active columns** and $\mathcal{A}$ will be referred to as the **active set**.

D. Working assumptions and remarks

We assume that matrix $\boldsymbol{A}$ satisfies the unique representation property (URP), which is stronger than the full rank assumption [16].

Definition 1. *A matrix* $\boldsymbol{A}$ *of size* $m \times n$ *(*$m \leqslant n$*) satisfies the URP if any selection of* m *columns of* $\boldsymbol{A}$ *forms a family of linearly independent vectors. By extension, a matrix* $\boldsymbol{A}$ *of size* $m \times n$ *(*$m > n$*) will be referred to as URP if it is full rank.*

Lemma 1. *Given a subset* $\mathcal{A}$ *of* $\{1, \ldots, n\}$ *whose cardinality is smaller than* $\min(m, n)$ *and provided that* $\boldsymbol{A}$ *satisfies the URP, the constrained least-square problem* (1) *has a unique minimizer, denoted by:*

$$\boldsymbol{x}_{\mathcal{A}} = \arg\min_{\{\boldsymbol{x} | \mathcal{A}(\boldsymbol{x}) \subseteq \mathcal{A}\}} \mathcal{E}(\boldsymbol{x}). \tag{3}$$

Proof: Let us denote by $\boldsymbol{A}_{\mathcal{A}}$ the matrix formed of all the active columns of $\boldsymbol{A}$ and by $\boldsymbol{z}$ the corresponding vector extracted from $\boldsymbol{x}$. Then, (1) is equivalent to the unconstrained minimization of $\|\boldsymbol{y} - \boldsymbol{A}_{\mathcal{A}}\boldsymbol{z}\|^2$ with respect to $\boldsymbol{z}$. Due to the URP, the columns of $\boldsymbol{A}_{\mathcal{A}}$ form an independent family of vectors, thus the unconstrained minimization of $\|\boldsymbol{y} - \boldsymbol{A}_{\mathcal{A}}\boldsymbol{z}\|^2$ with respect to $\boldsymbol{z}$ yields a unique minimizer. ☐

The values of λ and $\mathcal{A}$ being given, let us now consider the minimization problem (2). The search for a minimizer of $\mathcal{J}(\boldsymbol{x}; \lambda)$ on the domain $\{\boldsymbol{x} | \mathcal{A}(\boldsymbol{x}) \subseteq \mathcal{A}\}$ would require to compute $\mathcal{J}(\boldsymbol{x}_{\mathcal{A}'}; \lambda)$ for all the subsets $\mathcal{A}' \subseteq \mathcal{A}$. This is why vector $\boldsymbol{x}_{\mathcal{A}}$ cannot be guaranteed to be an optimizer to (2). In the following, we will denote by

$$\mathcal{J}_{\mathcal{A}}(\lambda) \triangleq \mathcal{J}(\boldsymbol{x}_{\mathcal{A}}; \lambda) = \mathcal{E}_{\mathcal{A}} + \lambda\|\boldsymbol{x}_{\mathcal{A}}\|_0. \tag{4}$$

the penalized cost function evaluated at $\boldsymbol{x}_{\mathcal{A}}$.

Remark 2. *The non uniqueness of the minimizers of* (L2L0C) *and* (L2L0P) *is due to the fact that two distinct subsets* $\mathcal{A}$ *and* $\mathcal{A}'$ *of* $\{1, \ldots, n\}$ *may yield the same least-square errors* $\mathcal{E}_{\mathcal{A}} = \mathcal{E}'_{\mathcal{A}}$. *However, it can be shown [14] that the minimizers of* (L2L0C) *and* (L2L0P) *necessarily take the form* $\boldsymbol{x}_{\mathcal{A}}$, *and because the number of possible subsets* $\mathcal{A}$ *of* $\{1, \ldots, n\}$ *is finite, the sets* $\mathcal{X}_c(k)$ *and* $\mathcal{X}_p(\lambda)$ *are of finite cardinality, provided that* $k \leqslant \min(m, n)$ *and that* $\lambda > 0$, *respectively.*

III. SBR algorithm for L2L0P at fixed λ

In this section, we develop a heuristic algorithm to estimate the solution of (L2L0P) for a fixed hyperparameter value λ. This algorithm is inspired by the SMLR algorithm [11–13]. Let us first recall its principle and the corresponding Bayesian background.

A. SMLR algorithm

A sparse signal $\boldsymbol{x}$ can be modeled as a Bernoulli-Gaussian random vector; each sample of $\boldsymbol{x}$ reads $x_i = q_i r_i$, where the Bernoulli random variable $q_i \sim \mathcal{B}(\rho)$ is coding the presence ($q_i = 1$) or absence ($q_i = 0$) of signal at location i and ρ is the probability of presence of signal at location i. The amplitude r_i is a Gaussian random variable whose distribution is $r_i | q_i \sim \mathcal{N}(0, q_i \sigma_x^2)$. Note that the set of locations i such that $q_i = 1$ coincides with the notion of active set introduced in Section II.

For inverse problems of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{A}$ is the observation matrix and $\boldsymbol{n}$ stands for the observation noise, sparse signal approximation can be done by first, maximizing the *posterior* likelihood of $\boldsymbol{q} = [q_1, \ldots, q_n]^T$ and then, by deducing the amplitudes $\boldsymbol{r}$ given the knowledge of $\boldsymbol{q}$ [12, 13]. In [13], it is shown that provided that the noise $\boldsymbol{n}$ is Gaussian and independent from $\boldsymbol{x}$, the *posterior* log-likelihood of $\boldsymbol{q}$ reads:

$$L(\boldsymbol{q}|\boldsymbol{y}) \propto -\boldsymbol{y}^T \boldsymbol{B}_{\boldsymbol{q}}^{-1} \boldsymbol{y} - \log |\boldsymbol{B}_{\boldsymbol{q}}| - 2 \log\left(\frac{1}{\rho} - 1\right) \|\boldsymbol{q}\|_0 \quad (5)$$

up to an additional constant, where $\boldsymbol{B}_{\boldsymbol{q}}$ is a matrix depending on $\boldsymbol{A}$ and $\boldsymbol{q}$, and symbol $\propto$ indicates proportionality.

The SMLR algorithm is a coordinate-wise ascent algorithm to maximize $L(\boldsymbol{q}|\boldsymbol{y})$ with respect to $\boldsymbol{q}$. At each iteration, all possible single replacements of $\boldsymbol{q}$ (set $q_i = 1 - q_i$ while keeping the other $q_j, j \neq i$ unchanged) are tested, then the "most likely" replacement is chosen, *i.e.*, the one yielding the largest increase of $L(\boldsymbol{q}|\boldsymbol{y})$. This task is repeated iteratively until no single replacement can increase $L(\boldsymbol{q}|\boldsymbol{y})$ anymore.

Interestingly, the form of the *posterior* likelihood (5) shares similarities with the (L2L0P) cost function in that the last term involves the L0-norm of $\boldsymbol{q}$. This similarity motivates the development of our proposed algorithm, referred to as Single Best Replacement (SBR) to distinguish it from SMLR and to remove the statistical connotation which is not necessarily appropriate in the L2-L0 minimization context.

B. SBR algorithm

SBR is an iterative search algorithm to solve (L2L0P) for fixed λ. At each iteration, the n possible single replacements are tested, then the best one is chosen, allowing $\boldsymbol{x}$ to be updated. The word replacement includes two cases: either one single non-active sample ($q_i = 0$) is selected to become active or one single active sample ($q_i = 1$) is selected to become non-active. For convenience, we define

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

to refer to the addition ($\cup$) or deletion ($\backslash$) of an index i into/from an active set $\mathcal{A}$.

Remark 3. *For a given active set $\mathcal{A}$, the corresponding least-square solution is the vector $\boldsymbol{x}_{\mathcal{A}}$ defined in* (3). *Therefore, for the SBR algorithm and its further extensions below, with a slight abuse of words, we will consider $\mathcal{A}$ as a current iterate, although strictly speaking, the current iterate is the vector $\boldsymbol{x}_{\mathcal{A}}$ derived from $\mathcal{A}$.*

Let us assume that the current active set $\mathcal{A}$ is given. For all indices $i \in \{1, \ldots, n\}$, we compute the minimizer $\boldsymbol{x}_{\mathcal{A} \bullet i}$ of $\mathcal{E}(\boldsymbol{x})$ whose support is included in $\mathcal{A} \bullet i$, and we keep in memory the value of $\mathcal{J}_{\mathcal{A} \bullet i}(\lambda)$. Once all the possible single replacements have been tested, we search for the best one:

$$l = \arg\min_i \ \mathcal{J}_{\mathcal{A} \bullet i}(\lambda). \quad (7)$$

If $\mathcal{J}_{\mathcal{A} \bullet l}(\lambda)$ is strictly lower than $\mathcal{J}_{\mathcal{A}}(\lambda)$, we update $\mathcal{A}$ and $\boldsymbol{x}$: $\mathcal{A} \leftarrow \mathcal{A} \bullet l$ and $\boldsymbol{x}_{\mathcal{A}} \leftarrow \boldsymbol{x}_{\mathcal{A} \bullet l}$. This task is repeated until none of the single replacements $\mathcal{A} \bullet i$ yield a decrease of $\mathcal{J}_{\mathcal{A}}(\lambda)$.

Prior to any iteration, the SBR algorithm necessitates to define an initial active set, *e.g.*, the empty active set. SBR terminates after a finite number of iterations, because there exists a finite number (2^n) of possibilities for $\mathcal{A}$, and SBR is a descent algorithm.

C. Implementation and practical issues

At a given iteration k, SBR explores the n supports $\mathcal{A}_k \bullet i$ for all $i \in \{1, \ldots, n\}$ and computes the corresponding costs $\mathcal{E}_{\mathcal{A}_k \bullet i}$ and $\mathcal{J}_{\mathcal{A}_k \bullet i}(\lambda)$. Therefore, one SBR iteration requires the resolution of n least-square problems of the form (1). This computation can be very expensive when n is large, since n linear systems have to be solved, each normal matrix reading $\boldsymbol{A}_{\mathcal{A}}^T \boldsymbol{A}_{\mathcal{A}}$ where $\boldsymbol{A}_{\mathcal{A}}$ denotes the matrix made of the active columns of $\boldsymbol{A}$. Instead, we use a fast strategy based on the block matrix inversion lemma [17] to update $(\boldsymbol{A}_{\mathcal{A}}^T \boldsymbol{A}_{\mathcal{A}})^{-1}$ when $\mathcal{A}$ is modified. Similar ideas appeared in [12] and in [8] for speeding up SMLR and OMP/OLS, respectively.

IV. CONTINUATION ALGORITHM

The SBR algorithm works for a fixed λ-value. In this section, we propose an extension of SBR which estimates a "solution path" $\{\boldsymbol{x}_p(\lambda) | \lambda \geqslant 0\}$, *i.e.*, a sequence of estimates $\boldsymbol{x}_p(\lambda)$ for any sparsity level λ. In other words, for all λ, $\boldsymbol{x}_p(\lambda)$ is an estimate of an element of the set $\mathcal{X}_p(\lambda)$ defined in (L2L0P). The extended algorithm is named "Continuation Single Best Replacement" (CSBR), as only a finite number of λ-values are required to estimate a whole solution path. Finally, up to a slight adaptation of CSBR, a solution path $\{\boldsymbol{x}_c(k) | k \geqslant 0\}$ of the constrained L2-L0 problem (L2L0C) can be estimated.

A. CSBR algorithm

As mentioned in Section II, the dependence of the sets $\mathcal{X}_p(\lambda)$ with respect to λ ($\lambda \geqslant 0$) is piecewise constant, with a finite number of intervals $(\lambda_i^\star, \lambda_{i+1}^\star)$. Therefore, it is sufficient to search for these critical values $\lambda_i^\star$ for which the content of $\mathcal{X}_p(\lambda)$ is changing. The CSBR algorithm is a heuristic algorithm inspired by this piecewise constant property. The structure of the algorithm is the following:

1) estimate iteratively the critical λ-values (denoted by λ_q, where q is the iteration index, to distinguish the estimates with the exact critical values $\lambda_i^\star$);

2) for each value of λ_q, run SBR to estimate the corresponding active set $\mathcal{A}_q$.

We choose to start the algorithm at $\lambda_0 = +\infty$ ($\mathcal{A}_0$ is set to the empty set) and then to gradually decrease the λ-value.

For a given λ-value $\lambda = \lambda_q$, let $\mathcal{A}_q$ be the output of SBR. When SBR terminates, any replacement of $\mathcal{A}_q$ by $\mathcal{A}_q \bullet i$ yields an increase of the cost $\mathcal{J}_{\mathcal{A}_q}(\lambda)$:

$$\forall i \in \{1, \ldots, n\},\ \mathcal{J}_{\mathcal{A}_q \bullet i}(\lambda) \geqslant \mathcal{J}_{\mathcal{A}_q}(\lambda). \tag{8}$$

Because (8) holds for $\lambda = \lambda_q$, we search for the next λ-value λ_{q+1} ($\lambda_{q+1} < \lambda_q$) below which (8) does not hold anymore. λ_{q+1} is defined such that, when $\lambda < \lambda_{q+1}$, at least one index i yields a decrease of the cost function, *i.e.*, $\mathcal{J}_{\mathcal{A}_q \bullet i}(\lambda) < \mathcal{J}_{\mathcal{A}_q}(\lambda)$.

It can be shown [14] that for $\lambda < \lambda_q$, such a decrease can only occur when $\bullet$ is the union operation ($\bullet = \cup$). Up to a few manipulations of (8) and (4), the value of λ_{q+1} reads:

$$\lambda_{q+1} = \max_{i \notin \mathcal{A}_q} \{\mathcal{E}_{\mathcal{A}_q} - \mathcal{E}_{\mathcal{A}_q \cup i}\}, \tag{9}$$

or $\lambda_{q+1} = 0$ when $\mathcal{A}_q$ is the complete set $\{1, \ldots, n\}$.

The CSBR algorithm stops when $\lambda_{q+1} = 0$. In this case, the definition (9) implies that for all $i \notin \mathcal{A}_q$, $\mathcal{E}_{\mathcal{A}_q \cup i} = \mathcal{E}_{\mathcal{A}_q}$. Finally, and because of the definition of $\boldsymbol{x}_{\mathcal{A}_q}$ in (3), $\boldsymbol{x}_{\mathcal{A}_q}$ is necessarily an unconstrained least-square estimate.

B. Adaptation of CSBR to the constrained L2-L0 problem

We can straightforwardly adapt CSBR to provide an estimation of a "solution path" $\{\boldsymbol{x}_c(k) | k \geqslant 0\}$ of the constrained L2-L0 minimization problem (L2L0C), up to a storage of the intermediate SBR iterates.

Indeed, at the beginning of CSBR, the initial active set is $\mathcal{A}_0 = \emptyset$ and during the SBR iterations, the support of the active set is only modified by one replacement (addition or deletion) at a time. Therefore, the sequence of the SBR iterates provides at least one estimate whose L0-norm is equal to k for all $k = 0, \ldots, K$, where K is the maximum of the L0-norms of the SBR iterates explored while running CSBR. Estimating a solution path $\{\boldsymbol{x}_c(k) | k = 0, \ldots, K\}$ can finally be done by storing for each value of k, the best SBR iterate $\boldsymbol{x}_{\mathcal{A}}$ whose L0-norm is equal to k, *i.e.*, the iterate having k non-zero entries and yielding the lowest residual error $\mathcal{E}_{\mathcal{A}}$.

V. SIMULATION RESULTS

In this section, we focus on the constrained L2-L0 problem. We compare the adapted CSBR algorithm with two other algorithms providing an estimation of a solution path $\{\boldsymbol{x}_c(k) | k \geqslant 0\}$: Orthogonal Matching Pursuit (OMP) [8] and Orthogonal Least Squares (OLS) [9, 10]. The comparison is done on a sparse signal deconvolution problem [11].

A. Data simulation for sparse signal deconvolution

The deconvolution problem consists in the estimation of a sparse signal $\boldsymbol{x}^*$ from a convoluted and noisy observation $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}^* + \boldsymbol{n}$, where $\boldsymbol{h}$ is the impulse response yielding a Toeplitz observation matrix $\boldsymbol{A}$. The deconvolution problem is

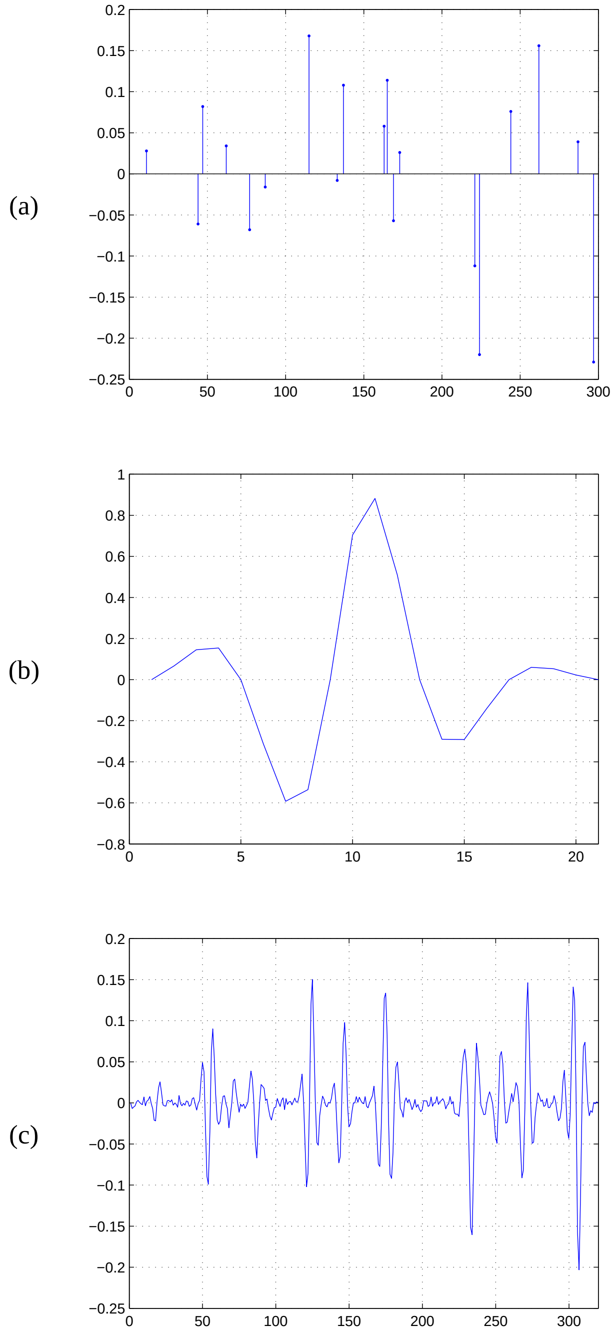
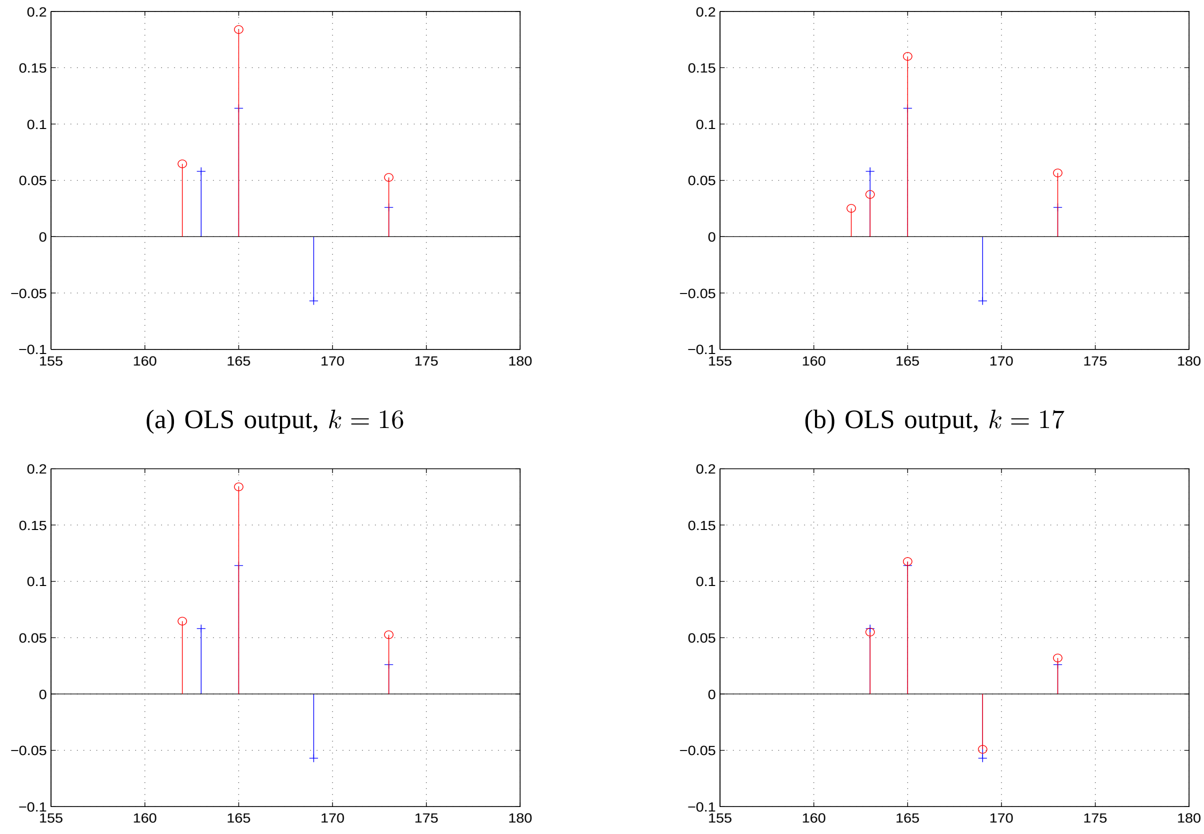


Fig. 1. Sparse signal deconvolution: data simulation. (a) unknown "true" sparse signal $\boldsymbol{x}^*$, including $\|\boldsymbol{x}^*\|_0 = 19$ impulses. (b) Impulse response $\boldsymbol{h}$. (c) Observation $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}^* + \boldsymbol{n}$ (the SNR is set to 20 dB).

particularly difficult when several impulses in $\boldsymbol{x}^*$ are located at neighboring locations (*e.g.*, x_i^* and $x_{i+1}^* \neq 0$, or x_i^* and $x_{i+2}^* \neq 0$), because the convolution $\boldsymbol{h} * \boldsymbol{x}^*$ results in a strong overlap of the responses to successive impulses. The simulated data designed in [11] are an illustration of such situation, in which $\boldsymbol{x}^*$ contains several close impulses (see Fig. 1 (a)). Knowing $\boldsymbol{x}^*$, the data generation consists of the computation



(a) OLS output, $k = 16$

(b) OLS output, $k = 17$

(c) CSBR output, $k = 16$

(d) CSBR output, $k = 17$

Fig. 2. OLS (a,b) and CSBR (c,d) recoveries of $\boldsymbol{x}^*$ for the observation $\boldsymbol{y}$ shown in Fig. 1 (c). The true signal $\boldsymbol{x}^*$ and its recoveries are represented in (blue) plus and (red) circles, respectively. Recoveries $\boldsymbol{x}_c(k)$ are represented for specific k-values ($k = 16$ and 17). For convenience, only zooms of the signals are shown in an area where several impulses occur in $\boldsymbol{x}^*$. (a,b) OLS consecutive recoveries. (c,d) CSBR recoveries.

of $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}^* + \boldsymbol{n}$. The signal to noise ratio (SNR), defined by $\text{SNR} = 10 \log_{10}(v_x/v_n)$, where v_x and v_n are the respective variances of the noiseless signal $\boldsymbol{x}^*$ and the noise process $\boldsymbol{n}$, is set to 20 dB. The signals $\boldsymbol{x}^*$, $\boldsymbol{h}$ and $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}^* + \boldsymbol{n}$ are shown in Fig. 1 (a,b,c).

The recoveries $\boldsymbol{x}_c(k)$ of $\boldsymbol{x}^*$ with OLS and CSBR are illustrated in Fig. 2. For each algorithm, two consecutive outputs are shown, for $k = 16$ and 17. After iteration # 16 of OLS, the entry $i = 162$ has been included into the active set. This impulse detection is false, since $\boldsymbol{x}^*$ contains two impulses at entries 163 and 165 but none at $i = 162$. After iteration # 17, $i = 162$ remains in the active set because OLS only performs additions into the active set. As for CSBR, the same false detection occurs in $\boldsymbol{x}_c(16)$ but for the next output $\boldsymbol{x}_c(17)$, the entry $i = 162$ is deleted from the active set while two other true entries are included.

B. Comparative study and robustness analysis

The following simulations aim at evaluating the average performance and the robustness of OMP, OLS and CSBR. The notion of robustness is related to:

1) the level of noise embedded in the data;
2) the relative locations of the impulses in $\boldsymbol{x}^*$.

For this reason, we design a collection of data sets at different noise levels and for each noise level, we generate a number of observations $\boldsymbol{y}^j$ corresponding to random values $\boldsymbol{x}^{j*}$ of $\boldsymbol{x}^*$ and random noise realizations $\boldsymbol{n}^j$. The impulse response $\boldsymbol{h}$ is kept constant during the simulations (see Fig. 1 (b)).

We generate simulated data at 13 noise levels varying from 0 to 60 dB. At each SNR, 5000 Monte Carlo experiences are carried out, in which $\boldsymbol{x}^*$ is sampled using the Bernoulli-Gaussian model introduced in Section III. The Bernoulli parameter is set to 0.05 (*i.e.*, about 5 % of the samples of $\boldsymbol{x}^*$ are not equal to zero) and the variance of the amplitudes is set to 0.01. Each data generation yields a couple of signals $\boldsymbol{x}^{j*}$ and $\boldsymbol{y}^j = \boldsymbol{h} * \boldsymbol{x}^{j*} + \boldsymbol{n}^j$.

At a given SNR, we run the three algorithms for each observation $\boldsymbol{y}^j$. OMP and OLS are run until the iteration $k = \|\boldsymbol{x}^{j*}\|_0$, since we want to compare the k-th output $\boldsymbol{x}_c(k)$ with the sparse signal $\boldsymbol{x}^{j*}$ to be recovered. Similarly, the execution of CSBR is stopped when an iterate $\mathcal{A}_q$ has a cardinality greater than $k+3$ and we select the best solution having k components exactly. For each algorithm, the solution $\boldsymbol{x}_c(k)$ is said successful if $\boldsymbol{x}_c(k)$ and $\boldsymbol{x}^{j*}$ share exactly the same support. The rates of successful approximations are evaluated for each algorithm and for each SNR (see Fig. 3).

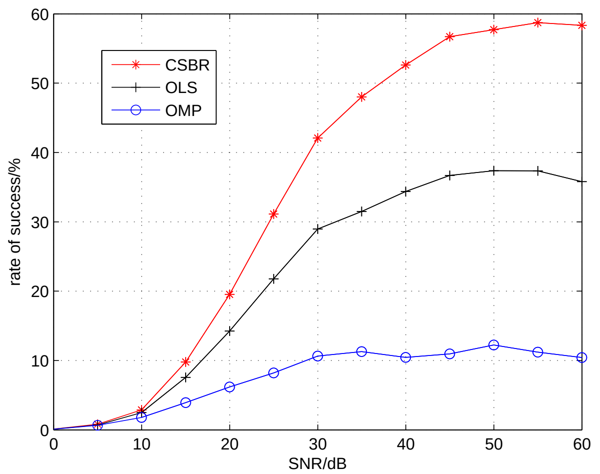


Fig. 3. Deconvolution problem: robustness of OMP, OLS and CSBR. The rate of successful approximations is displayed as a function of the SNR for each algorithm. The rates are evaluated from 5000 approximations, *i.e.*, while processing 5000 simulated observations $\boldsymbol{y}^j$ with each algorithm.

On average, the CSBR algorithm yields better results than OMP and OLS in that successful approximations occur more frequently. This is in coherence with our expectations because CSBR allows both additions and deletions while OMP and OLS do not. Once a wrong index is included into the active set, neither OMP nor OLS can delete it in the further iterations, thus disabling a successful approximation. For low levels of noise, the rates of successful approximations are poor and similar, as shown in Fig. 3. This rates need to be commented because they do not represent the true behavior of the three algorithms. They can mainly be justified by the binary criterion used for successful approximation (*i.e.*, the *exact* estimation of the unknown support). A low rate does not necessarily mean that the quality of approximation is poor in a qualitative viewpoint. Actually, the CSBR algorithm performs better than OMP and OLS even for low SNRs, although it rarely finds the exact expected support. A comparison of the least-square costs of the three estimates $\boldsymbol{x}_c(k)$ shows that for each k-value and for each SNR, OMP and OLS almost never find a better solution than CSBR while the contrary is very often true.

The computation burden depends on the size of the data and the number of iterations after which the algorithms are stopped. For the simulations described above, the observation matrix is of size 320×300 and we ran the algorithms until $k = \|\boldsymbol{x}^{j*}\|_0$ for OMP and OLS, and $k = \|\boldsymbol{x}^{j*}\|_0+3$ for CSBR. On average, $\|\boldsymbol{x}^{j*}\|_0$ is equal to 15, *i.e.*, 5 % of the 300 samples of $\boldsymbol{x}^{j*}$, and the average computation time of OMP, OLS and CSBR amount to 14, 52, and 245 milliseconds, respectively.

VI. Conclusion

Our main contribution was to propose an iterative algorithm to estimate a solution path of the L0-constrained least-square problem. Rather than starting from the constrained formulation, we addressed the L0-penalized least-square problem because the L0 penalization term allows the addition and/or deletion of indices into/from the solution support resulting in the so-called SBR algorithm. Then, this algorithm was extended to a continuation version yielding an estimation of a whole solution path of the penalized problem. Up to a slight adaptation, this algorithm also provides a solution path estimate of the constrained problem.

For particular dictionaries, namely, orthogonal dictionaries, the simple and fast OMP algorithm provides an optimal solution path. However, in more difficult problems such as sparse signal deconvolution, the proposed algorithm performs significantly better than OMP and OLS at the price of an increased computational burden. In such cases, the choice of the algorithm depends on the desired quality of approximation and on the available time.

References

[1] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries", *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[2] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations", *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, Mar. 1997.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *Annals Statist.*, vol. 32, no. 2, pp. 407–451, 2004.

[5] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation", in *Proc. IEEE ICASSP*, Philadephia, PA, Mar. 2005, vol. V, pp. 733–736.

[6] J.-J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise", *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3601–3608, Oct. 2005.

[7] D. L. Donoho and Y. Tsaig, "Fast solution of l_1-norm minimization problems when the solution may be sparse", Research Report, Stanford University, Stanford, CA, 2006.

[8] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, vol. 1, pp. 40–44.

[9] 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.

[10] T. Blumensath and M. E. Davies, "On the difference between orthogonal matching pursuit and orthogonal least squares", Tech. Rep., University of Edinburgh, Mar. 2007.

[11] J. J. Kormylo and J. M. Mendel, "Maximum-likelihood detection and estimation of Bernoulli-Gaussian processes", *IEEE Trans. Inf. Theory*, vol. 28, pp. 482–488, 1982.

[12] Y. Goussard, G. Demoment, and J. Idier, "A new algorithm for iterative deconvolution of sparse spike trains", in *Proc. IEEE ICASSP*, Albuquerque, NM, Apr. 1990, pp. 1547–1550.

[13] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation", *IEEE Trans. Signal Processing*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[14] J. Duan, C. Soussen, D. Brie, and J. Idier, "On the properties of the solution path of the constrained and penalized L2-L0 problems", Tech. Rep., Centre de Recherche en Automatique de Nancy, Jan. 2009.

[15] 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, 2007.

[16] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm", *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[17] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, Third edition, 1996.