From Bernoulli–Gaussian Deconvolution to Sparse Signal Restoration

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

***Abstract*—Formulated as a least square problem under an ℓ_0 constraint, sparse signal restoration is a discrete optimization problem, known to be NP complete. Classical algorithms include, by increasing cost and efficiency, matching pursuit (MP), orthogonal matching pursuit (OMP), orthogonal least squares (OLS), stepwise regression algorithms and the exhaustive search. We revisit the single most likely replacement (SMLR) algorithm, developed in the mid-1980s for Bernoulli–Gaussian signal restoration. We show that the formulation of sparse signal restoration as a limit case of Bernoulli–Gaussian signal restoration leads to an ℓ_0-penalized least square minimization problem, to which SMLR can be straightforwardly adapted. The resulting algorithm, called single best replacement (SBR), can be interpreted as a forward–backward extension of OLS sharing similarities with stepwise regression algorithms. Some structural properties of SBR are put forward. A fast and stable implementation is proposed. The approach is illustrated on two inverse problems involving highly correlated dictionaries. We show that SBR is very competitive with popular sparse algorithms in terms of tradeoff between accuracy and computation time.**

***Index Terms*—Bernoulli-Gaussian (BG) signal restoration, inverse problems, mixed ℓ_2-ℓ_0 criterion minimization, orthogonal least squares, SMLR algorithm, sparse signal estimation, stepwise regression algorithms.**

I. Introduction

Sparse signal restoration arises in inverse problems such as Fourier synthesis, mono- and multidimensional deconvolution, and statistical regression. It consists in the decomposition of a signal $\boldsymbol{y}$ as a linear combination of a limited number of elements from a dictionary $\boldsymbol{A}$. While formally very similar, sparse signal restoration has to be distinguished from sparse signal approximation. In sparse signal restoration, the choice of the dictionary is imposed by the inverse problem at hand whereas in sparse approximation, the dictionary has to be chosen according to its ability to represent the data with a limited number of coefficients.

Sparse signal restoration can be formulated as the minimization of the squared error $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ (where $\|\cdot\|$ refers to the Euclidean norm) under the constraint that the ℓ_0 pseudo-norm of $\boldsymbol{x}$, defined as the number of nonzero entries in $\boldsymbol{x}$, is small. This problem is often referred to as subset selection because it consists in selecting a subset of columns of $\boldsymbol{A}$. This yields a discrete problem (since there are a finite number of possible subsets) which is known to be NP-complete [1]. In this paper, we focus on "difficult" situations in which some of the columns of $\boldsymbol{A}$ are highly correlated, the unknown weight vector $\boldsymbol{x}$ is only approximately sparse, and/or the data are noisy. To address subset selection in a fast and suboptimal manner, two approaches can be distinguished.

The first one, which has been the most popular in the last decade, approximates the subset selection problem by a continuous optimization problem, convex or not, that is easier to solve [2]–[7]. In particular, the ℓ_1 relaxation of the ℓ_0-norm has been increasingly investigated [2], [3], leading to the LASSO optimization problem.

The second approach addresses the *exact* subset selection problem using either iterative thresholding [8]–[11] or greedy search algorithms. The latter gradually increase or decrease by one the set of active columns. The simplest greedy algorithms are matching pursuit (MP) [12] and the improved version orthogonal matching pursuit (OMP) [13]. Both are referred to as forward greedy algorithms since they start from the empty active set and then gradually increase it by one element. In contrast, the backward algorithm of Couvreur and Bresler [14] starts from a complete active set which is gradually decreased by one element. It is, however, only valid for undercomplete dictionaries. Forward–backward algorithms (also known as stepwise regression algorithms) in which insertions and removals of dictionary elements are both allowed, are known to yield better recovery performance since an early wrong selection can be counteracted by its further removal from the active set [15]–[18]. In contrast, the insertion of a wrong element is irreversible when using forward algorithms. We refer the reader to [18, Ch. 3] for an overview of the forward–backward algorithms in subset selection.

The choice of the algorithm depends on the amount of time available and on the structure of matrix $\boldsymbol{A}$. In favorable cases, the

Manuscript received January 14, 2011; revised May 13, 2011; accepted June 02, 2011. Date of publication June 27, 2011; date of current version September 14, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark Coates. This work was carried out in part while C. Soussen was visiting IRCCyN during the academic year 2010–2011 with the financial support of CNRS.

C. Soussen and D. Brie are with the Centre de Recherche en Automatique de Nancy, CRAN, UMR 7039, Nancy-University, CNRS, F-54506 Vandœuvre-lès-Nancy, France (e-mail: Charles.Soussen@cran.uhp-nancy.fr; David.Brie@cran.uhp-nancy.fr).

J. Idier is with the Institut de Recherche en Communications et Cybernétique de Nantes, IRCCyN, UMR CNRS 6597, F-44321 Nantes, France (e-mail: Jerome.Idier@irccyn.ec-nantes.fr).

J. Duan was with CRAN. He now is with the Department of Biomedical Engineering and Biostatistics, Tulane University, New Orleans, LA 70112 USA (e-mail: jduan@tulane.edu).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes the basic Matlab function implementing the Single Best Replacement (SBR) algorithm together with a simple test program showing how to call the SBR function. This material is 268 KB in size.

Digital Object Identifier 10.1109/TSP.2011.2160633

1053-587X/$26.00 © 2011 IEEE

suboptimal search algorithms belonging to the first or the second approach provide solutions having the same support as the exhaustive search solution. Specifically, if the unknown signal is highly sparse and if the correlation between any pair of columns of $\boldsymbol{A}$ is low, the ℓ_1-norm approximation provides optimal solutions [3]. But when fast algorithms are unsatisfactory, it is relevant to consider slower algorithms being more accurate and remaining very fast compared to the exhaustive search. The orthogonal least squares algorithm (OLS) [19] which is sometimes confused with OMP [20], falls into this category. Both OLS and OMP share the same structure, the difference being that at each iteration, OLS solves as many least square problems as there are nonactive columns while OMP only performs one linear inversion. In this paper, we derive a forward–backward extension of OLS allowing an insertion or a removal per iteration, each iteration requiring to solve n least square problems, where n is the size of $\boldsymbol{x}$.

The proposed forward–backward extension of OLS can be viewed as a new member of the family of stepwise regression algorithms. The latter family traces back to 1960 [15], and other popular algorithms were proposed in the 1980s [18] and more recently [21]. Note that forward–backward extensions of OMP have also been proposed [22], [23]. In contrast with the other stepwise regression algorithms, our approach relies on a bi-objective formulation in order to handle the tradeoff between low residual and low cardinality. This formulation reads as the minimization of the ℓ_0-penalized least square cost function $\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2+\lambda\|\boldsymbol{x}\|_0$. Then, we design a heuristic algorithm to minimize this cost function in a suboptimal way. While the other forward–backward strategies [15]–[17], [21], [22] aim at handling the same tradeoff, most of them are not expressed as optimization algorithms, but rather as empirical schemes without any connexion with an objective function. Moreover, some of them involve discrete search parameters that control variable selection or de-selection [15], [16], [22] while others do not involve any parameter [17], [21]. An exception can be made for Broersen's algorithm [17] since it aims at minimizing $\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2+\lambda\|\boldsymbol{x}\|_0$ for a specific λ value corresponding to Mallows' C_p statistic. However, it is only valid for undercomplete problems. On the contrary, our proposed algorithm is general and valid for any λ value. It does not necessitate to tune any other parameters (e.g., stopping parameters).

Our starting point is the single most likely replacement (SMLR) algorithm which proved to be a very efficient tool for the deconvolution of a Bernoulli–Gaussian (BG) signal [24]–[27]. We show that sparse signal restoration can be seen as a limit case of maximum *a posteriori* (MAP) BG restoration which results in an adaptation of SMLR to subset selection.

The paper is organized as follows. In Section II, we introduce the BG model and the Bayesian framework from which we formulate the sparse signal restoration problem. In Section III, we adapt SMLR resulting in the so-called single best replacement (SBR) algorithm. In Section IV, we propose a fast and stable SBR implementation. Finally, Sections V and VI illustrate the method on the sparse spike deconvolution with a Gaussian impulse response and on the joint detection of discontinuities at different orders in a signal.

II. Sparse Signal Estimation Using a Limit Bernoulli–Gaussian Model

A. Preliminary Definitions and Working Assumptions

Given an observation vector $\boldsymbol{y}\in\mathbb{R}^m$ and a dictionary $\boldsymbol{A}=[\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n]\in\mathbb{R}^{m\times n}$, a subset selection algorithm aims at computing a weight vector $\boldsymbol{x}\in\mathbb{R}^n$ yielding an accurate approximation $\boldsymbol{y}\approx\boldsymbol{A}\boldsymbol{x}$. The columns $\boldsymbol{a}_i$ corresponding to the nonzero weights x_i are referred to as the active (or selected) columns.

Throughout this paper, no assumption is made on the size of $\boldsymbol{A}$: m can be either smaller or larger than n. $\boldsymbol{A}$ is assumed to satisfy the unique representation property (URP): any $\min(m,n)$ columns of $\boldsymbol{A}$ are linearly independent. This assumption is usual when $m\leqslant n$; it is stronger than the full rank assumption [28]. When $m\geqslant n$, it amounts to the full rank assumption. Although URP was originally introduced to guarantee uniqueness of sparse solutions [28], we use this assumption to propose a valid algorithm. It can actually be relaxed provided that the search strategy guarantees that the selected columns are linearly independent (see Section VI-C for details).

The support of a vector $\boldsymbol{x}\in\mathbb{R}^n$ is the set $\mathcal{S}(\boldsymbol{x})\subseteq\{1,\ldots,n\}$ defined by $i\in\mathcal{S}(\boldsymbol{x})$ *if and only if* $x_i\neq 0$. We denote by $\mathcal{Q}\subseteq\{1,\ldots,n\}$ the active set and by $\boldsymbol{q}\in\{0,1\}^n$ the related vector defined by $q_i=1$ *if and only if* $i\in\mathcal{Q}$. When $\mathrm{Card}[\mathcal{Q}]\leqslant\min(m,n)$, let $\boldsymbol{A}_\mathcal{Q}$ be the submatrix of size $m\times\mathrm{Card}[\mathcal{Q}]$ formed of the active columns of $\boldsymbol{A}$. We define the least square solution and the related squared error:

$$\boldsymbol{x}_\mathcal{Q}\triangleq\underset{\mathcal{S}(\boldsymbol{x})\subseteq\mathcal{Q}}{\operatorname{argmin}}\{\mathcal{E}(\boldsymbol{x})=\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2\} \tag{1}$$

$$\mathcal{E}_\mathcal{Q}\triangleq\mathcal{E}(\boldsymbol{x}_\mathcal{Q})=\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_\mathcal{Q}\|^2. \tag{2}$$

B. Bayesian Formulation of Sparse Signal Restoration

We consider the restoration of a sparse signal $\boldsymbol{x}$ from a linear observation $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{n}$, where $\boldsymbol{n}$ stands for the observation noise. An acknowledged probabilistic model dedicated to sparse signals is the BG model [24], [25], [27]. For such model, deterministic optimization algorithms [27] and Markov chain Monte Carlo techniques [29] are used to compute the MAP and the posterior mean, respectively. Hereafter, we define the BG model and then consider its estimation in the joint MAP sense.

A BG process can be defined using a Bernoulli random vector $\boldsymbol{q}\in\{0,1\}^n$ coding for the support and a Gaussian random vector $\boldsymbol{r}\sim\mathcal{N}(\boldsymbol{0},\sigma_x^2\boldsymbol{I}_n)$, with $\boldsymbol{I}_n$ the identity matrix of size n. Each sample x_i of $\boldsymbol{x}$ is modeled as $x_i=q_ir_i$ [24], [25]. The Bernoulli parameter $\rho=\Pr(q_i=1)$ is the probability of presence of signal and σ_x^2 controls the variance of the nonzero amplitudes $x_i=r_i$. The Bayesian formulation consists in inferring $\boldsymbol{x}=(\boldsymbol{q},\boldsymbol{r})$ knowing $\boldsymbol{y}$. The MAP estimator can be obtained by maximizing the marginal likelihood $l(\boldsymbol{q}\,|\,\boldsymbol{y})$ [27] or the joint likelihood $l(\boldsymbol{q},\boldsymbol{r}\,|\,\boldsymbol{y})$ [25], [26]. Following [25] and assuming a Gaussian white noise $\boldsymbol{n}\sim\mathcal{N}(\boldsymbol{0},\sigma_n^2\boldsymbol{I}_m)$, independent from $\boldsymbol{x}$, Bayes' rule leads to

$$\begin{aligned}\mathcal{L}(\boldsymbol{q},\boldsymbol{r})&\triangleq-2\sigma_n^2\log[l(\boldsymbol{q},\boldsymbol{r}\,|\,\boldsymbol{y})]\\&=\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\Delta}_{\boldsymbol{q}}\boldsymbol{r}\|^2+\frac{\sigma_n^2}{\sigma_x^2}\|\boldsymbol{r}\|^2+\lambda\|\boldsymbol{q}\|_0+c\end{aligned} \tag{3}$$

where $\lambda = 2\sigma_n^2 \log(1/\rho - 1)$, $\boldsymbol{\Delta_q}$ is the diagonal matrix of size n whose diagonal elements are q_i ($\boldsymbol{x}$ reads $\boldsymbol{x} = \boldsymbol{\Delta_q r}$), and c is a constant.

Now, a signal $\boldsymbol{x}$ is sparse if some entries x_i are equal to 0. Since this definition does not impose constraints on the range of the nonzero amplitudes, we choose to use a limit BG model in which the amplitude variance σ_x^2 is set to infinity. Note that a parallel limit development was done, independently from our work, in the conference paper [23]. In Appendix A, we show that the minimization of $\mathcal{L}$ w.r.t. $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{r})$ rereads

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \{\mathcal{J}(\boldsymbol{x}; \lambda) = \|\boldsymbol{y} - \boldsymbol{Ax}\|^2 + \lambda \|\boldsymbol{x}\|_0\}. \tag{4}$$

This formulation is close to that obtained in the Bayesian subset selection literature [18, Ch. 7] using an alternative BG model. In the latter model, the Gaussian prior relies on $\boldsymbol{R}_\mathcal{Q}\boldsymbol{r}$ instead of $\boldsymbol{r}$, with $\boldsymbol{R}_\mathcal{Q}$ the Cholesky factor of the Gram matrix $\boldsymbol{A}_\mathcal{Q}^t \boldsymbol{A}_\mathcal{Q}$. This leads to a cost function of the form (4), the difference being that λ depends on the amplitude variance σ_x^2 and tends to infinity as σ_x^2 tends to infinity [30], [31].

Remark 1 (Noise-Free Case): The Bayesian development above is valid for noisy data. In the noise-free case, we define the sparse solution as the limit of $\arg\min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x}; \lambda)$ when λ tends towards 0. According to classical results in optimization [32, Ch. 17], if $\{\lambda_k\}$ is a sequence decreasing towards 0 and $\boldsymbol{x}_k$ is an exact global minimizer of $\mathcal{J}(\boldsymbol{x}; \lambda_k)$, then every limit point of the sequence $\{\boldsymbol{x}_k\}$ is a solution of $\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0$ s.t. $\|\boldsymbol{y} - \boldsymbol{Ax}\|^2$ is minimal. In Appendix B, we derive a more precise result: "the set of minimizers of $\mathcal{J}(\boldsymbol{x}; \lambda)$ is constant when λ is close enough to 0 ($\lambda \neq 0$). It is equal to the set of sparsest solutions to $\boldsymbol{y} = \boldsymbol{Ax}$ in the overcomplete case, and to the unconstrained least-squares solution in the undercomplete case."

In the following, we focus on the minimization problem (4). The hyperparameter λ is fixed. It controls the level of sparsity of the desired solution. The algorithm that will be developed relies on an efficient search of the support of $\boldsymbol{x}$. The search strategy is based on the definition of a neighborhood relationship between two supports: two supports are neighbors if one is nested inside the other and the largest support has one more element.

III. Single Best Replacement Algorithm

We propose to adapt the SMLR algorithm to the minimization of the mixed ℓ_2-ℓ_0 cost function $\mathcal{J}(\boldsymbol{x}; \lambda)$ defined in (4). To clearly distinguish SMLR which specifically aims at minimizing (3), the adapted algorithm will be termed as single best replacement (SBR).

A. Principle of SMLR and Main Notations

SMLR [24] is a deterministic coordinatewise ascent algorithm to maximize likelihood functions of the form $l(\boldsymbol{q} \,|\, \boldsymbol{y})$ (marginal MAP estimation) or $l(\boldsymbol{q}, \boldsymbol{r} \,|\, \boldsymbol{y})$ (joint MAP estimation). In the latter case, it is easy to check from (3) that given $\boldsymbol{q}$, the minimizer of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{r})$ w.r.t. $\boldsymbol{r}$ has a closed form expression $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$. Consequently, the joint MAP estimation reduces to the minimization of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{r}(\boldsymbol{q}))$ w.r.t. $\boldsymbol{q}$. At each SMLR iteration, all the possible single replacements of the support $\boldsymbol{q}$ (set $q_i = 1 - q_i$ while keeping the other q_j, $j \neq i$ unchanged) are tested, then the replacement yielding the maximal decrease of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{r}(\boldsymbol{q}))$ is chosen. This task is repeated until no single replacement can decrease $\mathcal{L}(\boldsymbol{q}, \boldsymbol{r}(\boldsymbol{q}))$ anymore. The number of possible supports $\boldsymbol{q}$ being finite and SMLR being a descent algorithm, it terminates after a finite number of iterations.

Before adapting SMLR, let us introduce some useful notations. We denote by $\mathcal{Q} \bullet i$ a single replacement, i.e., an insertion or removal into/from the active set $\mathcal{Q}$:

$$\mathcal{Q} \bullet i \triangleq \begin{cases} \mathcal{Q} \cup \{i\} & \text{if } i \notin \mathcal{Q} \\ \mathcal{Q} \backslash \{i\} & \text{otherwise.} \end{cases}$$

When $\mathrm{Card}[\mathcal{Q}] \leqslant \min(m, n)$, we define the cost function

$$\mathcal{J}_\mathcal{Q}(\lambda) \triangleq \mathcal{E}_\mathcal{Q} + \lambda \mathrm{Card}[\mathcal{Q}] \tag{5}$$

involving the squared error $\mathcal{E}_\mathcal{Q}$ defined in (2). By definition of $\mathcal{J}(\boldsymbol{x}_\mathcal{Q}; \lambda) = \mathcal{E}_\mathcal{Q} + \lambda \|\boldsymbol{x}_\mathcal{Q}\|_0$, $\mathcal{J}_\mathcal{Q}(\lambda)$ coincides with $\mathcal{J}(\boldsymbol{x}_\mathcal{Q}; \lambda)$ when the support of $\boldsymbol{x}_\mathcal{Q}$ is equal to $\mathcal{Q}$.

Although it aims at minimizing $\mathcal{J}(\boldsymbol{x}; \lambda)$, the proposed SBR algorithm involves the computation of $\mathcal{J}_\mathcal{Q}(\lambda)$ rather than $\mathcal{J}(\boldsymbol{x}_\mathcal{Q}; \lambda)$. We make this choice because $\mathcal{J}_\mathcal{Q}(\lambda)$ can be computed and updated more efficiently, the computation of $\boldsymbol{x}_\mathcal{Q}$ being no longer necessary. In Section III-C, we show that for noisy data, the replacement of $\mathcal{J}(\boldsymbol{x}_\mathcal{Q}; \lambda)$ by $\mathcal{J}_\mathcal{Q}(\lambda)$ has a negligible effect.

B. The Single Best Replacement Algorithm

SMLR can be seen as an exploration strategy for discrete optimization rather than an algorithm specific to a posterior likelihood function. Here, we use this strategy to minimize $\mathcal{J}(\boldsymbol{x}; \lambda)$. We rename the algorithm Single Best Replacement to remove any statistical connotation.

SBR works as follows. Consider the current support $\mathcal{Q}$. The n single replacements $\mathcal{Q} \bullet i$ are tested, i.e., we compute the squared errors $\mathcal{E}_{\mathcal{Q} \bullet i}$ and we memorize the values of $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$. If the minimum of $\mathcal{J}_{\mathcal{Q} \bullet i}(\lambda)$ is lower than $\mathcal{J}_\mathcal{Q}(\lambda)$, then we select the index yielding this minimum value:

$$\ell \in \arg\min_{i \in \{1, \ldots, n\}} \mathcal{J}_{\mathcal{Q} \bullet i}(\lambda). \tag{6}$$

The next SBR iterate is thus defined as $\mathcal{Q}' = \mathcal{Q} \bullet \ell$. This task is repeated until $\mathcal{J}_\mathcal{Q}(\lambda)$ cannot decrease anymore. By default, we use the initial empty support. The algorithm is summarized in Table I.

C. Case Where Some Active Amplitudes Are Zero

We show that this case almost surely never arises when the data $\boldsymbol{y}$ are corrupted with "nondegenerate" noise.

Theorem 1: Let $\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{n}$ where $\boldsymbol{y}_0 \in \mathbb{R}^m$ is fixed and $\boldsymbol{n}$ is an absolute continuous random vector, i.e., admitting a probability density w.r.t. the Lebesgue measure. Then, when $\mathrm{Card}[\mathcal{Q}] \leqslant \min(m, n)$, the probability that $\|\boldsymbol{x}_\mathcal{Q}\|_0 < \mathrm{Card}[\mathcal{Q}]$ is equal to 0.

Proof: Let $k = \mathrm{Card}[\mathcal{Q}]$ and $\boldsymbol{t}_\mathcal{Q}$ be the minimizer of $\|\boldsymbol{y} - \boldsymbol{A}_\mathcal{Q}\boldsymbol{t}\|^2$ over $\mathbb{R}^k$. $\boldsymbol{t}_\mathcal{Q}$ reads $\boldsymbol{t}_\mathcal{Q} = \boldsymbol{V}_\mathcal{Q}\boldsymbol{y}$ where matrix $\boldsymbol{V}_\mathcal{Q} = (\boldsymbol{A}_\mathcal{Q}^t \boldsymbol{A}_\mathcal{Q})^{-1} \boldsymbol{A}_\mathcal{Q}^t$ is of size $k \times m$, and $\|\boldsymbol{x}_\mathcal{Q}\|_0 = \|\boldsymbol{t}_\mathcal{Q}\|_0 \leqslant k$. Denoting by $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k \in \mathbb{R}^m$ the row vectors of $\boldsymbol{V}_\mathcal{Q}$, $\|\boldsymbol{t}_\mathcal{Q}\|_0 < k$ *if and only if* there exists i such that $\langle \boldsymbol{y}, \boldsymbol{v}^i \rangle = 0$ (where $\langle \cdot, \cdot \rangle$

TABLE 1
SBR ALGORITHM. BY DEFAULT, $\mathcal{Q}_1 = \emptyset$

```
Input: $\boldsymbol{A}$, $\boldsymbol{y}$, $\lambda$ and support $\mathcal{Q}_1$ ($\mathrm{Card}[\mathcal{Q}_1] \leqslant \min(m,n)$)
Step 1: Set $j = 1$.
Step 2: For $i \in \{1,\ldots,n\}$, compute $\mathcal{J}_{\mathcal{Q}_j \bullet i}(\lambda)$.
        Compute $\ell$ using (6).
        If $\mathcal{J}_{\mathcal{Q}_j \bullet \ell}(\lambda) < \mathcal{J}_{\mathcal{Q}_j}(\lambda)$,
            Set $\mathcal{Q}_{j+1} = \mathcal{Q}_j \bullet \ell$.
        else,
            Terminate SBR.
        End if.
        Set $j = j + 1$ and go to Step 2.
Output: support $\mathcal{Q}_j = \mathrm{SBR}(\mathcal{Q}_1; \lambda)$
```

denotes the inner product). Because $\boldsymbol{A}_\mathcal{Q}$ is full rank, $\boldsymbol{V}_\mathcal{Q}$ is full rank and then $\forall i$, $\boldsymbol{v}^i \neq \boldsymbol{0}$. Denoting by $\mathcal{H}^\perp(\boldsymbol{v}^i)$ the hyperplane of $\mathbb{R}^m$ which is orthogonal to $\boldsymbol{v}^i$, we have

$$\|\boldsymbol{x}_\mathcal{Q}\|_0 < k \Longleftrightarrow \boldsymbol{y} \in \bigcup_{i=1}^{k} \mathcal{H}^\perp(\boldsymbol{v}^i). \tag{7}$$

Because the set $\bigcup_i \mathcal{H}^\perp(\boldsymbol{v}^i)$ has a Lebesgue measure equal to zero and the random vector $\boldsymbol{y}$ admits a probability density, the probability of event (7) is zero. ■

Theorem 1 implies that when dealing with real noisy data, it is almost sure that all active coefficients x_i are nonzero. Hence, each SBR iterate $\mathcal{Q}$ almost surely satisfies $\mathcal{J}(\boldsymbol{x}_\mathcal{Q};\lambda) = \mathcal{J}_\mathcal{Q}(\lambda)$. In any case, SBR can be applied without restriction and the properties stated below (e.g., termination after a finite number of iterations) remain valid when an SBR iterate satisfies $\|\boldsymbol{x}_\mathcal{Q}\|_0 < \mathrm{Card}[\mathcal{Q}]$.

D. Properties of SBR

Proposition 1: Under the assumptions of Theorem 1, each SBR iterate $\boldsymbol{x}_\mathcal{Q}$ is almost surely a local minimizer of $\mathcal{J}(\boldsymbol{x};\lambda)$. In particular, the SBR output satisfies this property.

Proof: Let $\boldsymbol{x} = \boldsymbol{x}_\mathcal{Q}$ be an SBR iterate. According to Theorem 1, the support $\mathcal{S}(\boldsymbol{x}) = \mathcal{Q}$ almost surely. Setting $\varepsilon = \min_{i\in\mathcal{Q}} |x_i| > 0$, it is easy to check that if $\boldsymbol{x}' \in \mathbb{R}^n$ satisfies $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$, then $\mathcal{S}(\boldsymbol{x}') \supseteq \mathcal{S}(\boldsymbol{x}) = \mathcal{Q}$, thus $\|\boldsymbol{x}'\|_0 \geqslant \|\boldsymbol{x}\|_0$. Assume that $\boldsymbol{x}'$ satisfies $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$.

- If $\mathcal{S}(\boldsymbol{x}') = \mathcal{Q}$, then, by definition of $\boldsymbol{x} = \boldsymbol{x}_\mathcal{Q}$, we have $\mathcal{E}(\boldsymbol{x}') \geqslant \mathcal{E}(\boldsymbol{x})$. Thus, $\mathcal{J}(\boldsymbol{x}';\lambda) \geqslant \mathcal{J}(\boldsymbol{x};\lambda)$.
- Otherwise, $\mathcal{J}(\boldsymbol{x}';\lambda) = \mathcal{E}(\boldsymbol{x}') + \lambda\|\boldsymbol{x}'\|_0 \geqslant \mathcal{E}(\boldsymbol{x}') + \lambda(\|\boldsymbol{x}\|_0 + 1)$. By continuity of $\mathcal{E}$, there exists a neighborhood $\mathcal{V}(\boldsymbol{x})$ of $\boldsymbol{x}$ such that if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$, $|\mathcal{E}(\boldsymbol{x}') - \mathcal{E}(\boldsymbol{x})| < \lambda$. Thus, if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$, $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$ and $\mathcal{S}(\boldsymbol{x}') \supset \mathcal{Q}$, then $\mathcal{J}(\boldsymbol{x}';\lambda) > \mathcal{E}(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_0 = \mathcal{J}(\boldsymbol{x};\lambda)$.

Finally, if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$ and $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$, then $\mathcal{J}(\boldsymbol{x}';\lambda) \geqslant \mathcal{J}(\boldsymbol{x};\lambda)$. ■

Termination: Because SBR is a descent algorithm, a support $\mathcal{Q}$ cannot be explored twice and SBR terminates after a finite number of iterations. We emphasize that no stopping condition is needed unlike many algorithms which require to set a maximum number of iterations and/or a threshold on the squared error variation (CoSaMP, subspace pursuit, iterative hard thresholding, iterative reweighted ℓ_1).

OLS as a Special Case: When $\lambda = 0$, SBR coincides with the well known OLS algorithm [19], [33]. The removal operation never occurs because it yields an increase of the squared error $\mathcal{J}_\mathcal{Q}(0) = \mathcal{E}_\mathcal{Q}$.

Empty Solutions: We characterize the λ-values for which SBR yields an empty solution.

Remark 2: SBR $(\emptyset;\lambda)$ yields the empty set *if and only if* $\lambda \geqslant \lambda_{\max} \triangleq \max_i(\langle \boldsymbol{a}_i, \boldsymbol{y}\rangle^2/\|\boldsymbol{a}_i\|^2)$.

This result directly follows from checking that any insertion trial fails, i.e., $\forall i$, $\mathcal{E}_{\{i\}} + \lambda \geqslant \mathcal{E}_\emptyset$. It allows us to design an automatic procedure which sets a number of λ-values adaptively to the data in order to compute SBR solutions at different sparsity levels (see Section VI-D).

Relation Between SBR and SMLR: The main difference between both algorithms is that SMLR involves the inversion of a matrix of the form $\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{A}_\mathcal{Q} + \alpha\boldsymbol{I}_{\mathrm{Card}[\mathcal{Q}]}$ whereas SBR computes the inverse of $\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{A}_\mathcal{Q}$. In the case of SMLR, the term $\alpha\boldsymbol{I}_{\mathrm{Card}[\mathcal{Q}]}$ acts as a regularization on the amplitude values. It avoids instabilities when $\boldsymbol{A}_\mathcal{Q}$ is ill conditioned at the price of handling the additional hyperparameter α. On the contrary, instabilities may occur while using SBR. In the next section, we focus on this issue and propose a stable implementation.

IV. IMPLEMENTATION ISSUES

Given the current support $\mathcal{Q}$, an SBR iteration consists in computing the squared error $\mathcal{E}_{\mathcal{Q}'}$ for any replacement $\mathcal{Q}' = \mathcal{Q} \bullet i$, leading to the computation of $\mathcal{J}_{\mathcal{Q}'}(\lambda) = \mathcal{E}_{\mathcal{Q}'} + \lambda\mathrm{Card}[\mathcal{Q}']$. Our implementation is inspired by the fast implementation of the homotopy algorithm for ℓ_1 regression [3], [34]. It consists in maintaining the Cholesky factorization of the Gram matrix $\boldsymbol{G}_\mathcal{Q} \triangleq \boldsymbol{A}_\mathcal{Q}^t\boldsymbol{A}_\mathcal{Q}$ when $\mathcal{Q}$ is modified by one element. The Cholesky factorization takes the form $\boldsymbol{G}_\mathcal{Q} = \boldsymbol{L}_\mathcal{Q}\boldsymbol{L}_\mathcal{Q}^t$ where $\boldsymbol{L}_\mathcal{Q}$ is a lower triangular matrix of size $k = \mathrm{Card}[\mathcal{Q}]$. Also, $\boldsymbol{L}_\mathcal{Q}$ is better conditioned than $\boldsymbol{G}_\mathcal{Q}$, improving the stability of matrix inversion. We now give the main updating equations. Full detailed derivation can be found in Appendix C.

A. Efficient Strategy Based on the Cholesky Factorization

The replacement tests only rely on the current matrix $\boldsymbol{L}_\mathcal{Q}$ and do not require its update.

1) Single Replacement Tests: An insertion test $\mathcal{Q}' = \mathcal{Q} \cup \{i\}$ takes the form

$$\mathcal{J}_{\mathcal{Q}'}(\lambda) - \mathcal{J}_\mathcal{Q}(\lambda) = \lambda - \frac{\left(\boldsymbol{l}_{\mathcal{Q},i}^t\boldsymbol{L}_\mathcal{Q}^{-1}\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{y} - \boldsymbol{a}_i^t\boldsymbol{y}\right)^2}{\|\boldsymbol{a}_i\|^2 - \|\boldsymbol{l}_{\mathcal{Q},i}\|^2} \tag{8}$$

with $\boldsymbol{l}_{\mathcal{Q},i} = \boldsymbol{L}_\mathcal{Q}^{-1}\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{a}_i$. This computation mainly requires a triangular system inversion (computation of $\boldsymbol{l}_{\mathcal{Q},i}$ in $\mathcal{O}(k^2)$ elementary operations) up to the pre-computation of $\boldsymbol{L}_\mathcal{Q}^{-1}(\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{y})$ at the beginning of the current SBR iteration.

According to [18], [35], a removal test $\mathcal{Q}' = \mathcal{Q}\backslash\{i\}$ reads $\mathcal{J}_{\mathcal{Q}'}(\lambda) - \mathcal{J}_\mathcal{Q}(\lambda) = \boldsymbol{x}_\mathcal{Q}(i)^2/\gamma_i - \lambda$ where $\boldsymbol{x}_\mathcal{Q}(i)$ is the ith element in vector $\boldsymbol{x}_\mathcal{Q}$ and γ_i is the diagonal element of $\boldsymbol{G}_\mathcal{Q}^{-1}$ corresponding to the position of $\boldsymbol{a}_i$ in $\boldsymbol{A}_\mathcal{Q}$. The overall removal tests mainly amount to the inversion of the triangular matrix $\boldsymbol{L}_\mathcal{Q}$ (in $\mathcal{O}(k^3)$ operations) as the computation of γ_i for all i and of $\boldsymbol{G}_\mathcal{Q}^{-1}\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{y}$ (i.e., the values of $\boldsymbol{x}_\mathcal{Q}(i)$) from $\boldsymbol{L}_\mathcal{Q}^{-1}$ are both in $\mathcal{O}(k^2)$.

Note that insertion and removal tests can be easily done in parallel. In Matlab, this parallel implementation leads to a significant save of computation time due to the SIMD capabilities of Matlab.

2) Updating the Cholesky Factorization: The update of $\boldsymbol{L}_\mathcal{Q}$ can be easily done in the insertion case by adding the new column $\boldsymbol{a}_i$ at the last position in $\boldsymbol{A}_{\mathcal{Q}\cup\{i\}}$. The new matrix $\boldsymbol{L}_{\mathcal{Q}'}$ is a 2×2 block matrix whose upper left block is $\boldsymbol{L}_\mathcal{Q}$ (see Appendix C). The removal case requires more care since a removal breaks the triangular structure of $\boldsymbol{L}_\mathcal{Q}$. The update can be done by performing either a series of Givens planar rotations [21] or a positive rank 1 Cholesky update [36]. We describe the latter strategy in Appendix C. The Cholesky factorization update is in $\mathcal{O}(k^2)$ in the insertion case and in $\mathcal{O}((k-I)^2)$ in the removal case where I denotes the position of the column to be removed in $\boldsymbol{A}_\mathcal{Q}$.

B. Reduced Search

Additionally, we propose an acceleration of SBR yielding the same iterates with a reduced search. We notice that a column removal $\mathcal{Q}' = \mathcal{Q}\backslash\{i\}$ yields an increase of the squared error and a decrease of the penalty equal to λ. Hence, the maximum decrease of $\mathcal{J}_\mathcal{Q}(\lambda)$ which can be expected is λ. The acceleration of SBR consists in testing insertions first. If any insertion leads to $\mathcal{J}_\mathcal{Q}(\lambda) - \mathcal{J}_{\mathcal{Q}'}(\lambda) > \lambda$, then removals are not worth being tested. Otherwise, the removals have to be tested as stated in Table I. We have implemented this acceleration systematically.

C. Memory Requirements and Computation Burden

The actual implementation may vary depending on the size and the structure of matrix $\boldsymbol{A}$. We briefly describe the main possible implementations.

When the size of $\boldsymbol{A}$ is relatively small, the computation and storage of the Gram matrix $\boldsymbol{A}^t\boldsymbol{A}$ prior to any SBR iteration (storage of n^2 scalar elements) avoids to recompute the vectors $\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{a}_i$ which are needed when the insertion of $\boldsymbol{a}_i$ into the active set is tested. The storage of the other quantities (mainly $\boldsymbol{L}_\mathcal{Q}$) that are being updated amounts to $\mathcal{O}(k^2)$ scalar elements and a replacement test costs $\mathcal{O}(k^2)$ elementary operations in average.

When $\boldsymbol{A}$ is larger, the storage of $\boldsymbol{A}^t\boldsymbol{A}$ is no longer possible, thus $\boldsymbol{A}_\mathcal{Q}^t\boldsymbol{a}_i$ must be recomputed for any SBR iteration. This computation costs km elementary operations and now represents the most important part of an insertion test. When the dictionary has some specific structure, this limitation can be alleviated, enabling a fast implementation even for large n. For instance, if a large number of pairs of columns of $\boldsymbol{A}$ are orthogonal to each other, $\boldsymbol{A}^t\boldsymbol{A}$ can be stored as a sparse array. Also, finite impulse response deconvolution problems enable a fast implementation since $\boldsymbol{A}^t\boldsymbol{A}$ is then a Toeplitz matrix (save north-west and/or south-east submatrices, depending on the boundary conditions). The knowledge of the auto-correlation of the impulse response is sufficient to describe most of the Gram matrix.

All these variants have been implemented.[1] In the following, we analyze the behavior of SBR for two difficult problems involving highly correlated dictionaries: the deconvolution of a sparse signal with a Gaussian impulse response (Section V) and the joint detection of discontinuities at different orders in a signal (Section VI).

TABLE II
SEPARATION OF TWO GAUSSIAN FEATURES FROM NOISE-FREE DATA WITH SBR. d STANDS FOR THE DISTANCE BETWEEN THE GAUSSIAN FEATURES. WE DISPLAY THE SIZE OF THE SUPPORT OBTAINED FOR A SEQUENCE OF DECREASING λ-VALUES $\lambda_0 > \lambda_1 > \cdots > \lambda_7$. THE LABEL $\star$ INDICATES AN EXACT RECOVERY FOR A SUPPORT OF CARDINALITY 2

λ	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	$\lambda \leqslant \lambda_7$
$d = 20$	0	0	$2^\star$	$2^\star$	$2^\star$	$2^\star$	$2^\star$	$2^\star$
$d = 13$	0	1	3	4	5	$2^\star$	$2^\star$	$2^\star$
$d = 6$	0	1	1	3	5	6	8	$2^\star$

V. DECONVOLUTION OF A SPARSE SIGNAL WITH A GAUSSIAN IMPULSE RESPONSE

This is a typical problem for which SMLR was introduced [27]. It affords us to study the ability of SBR to perform an exact recovery in a simple noise-free case (separation of two Gaussian signals) and to test SBR in a noisy case (estimation of a larger number of Gaussians) and compare it with other algorithms. For simulated problems, we denote by $\boldsymbol{x}^\star$ the exact sparse signal, the data reading $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{n}$. The dictionary columns are always normalized: $\|\boldsymbol{a}_i\|^2 = 1$. The signal-to-noise ratio (SNR) is defined by $\mathrm{SNR} = 10\log(P_y/P_n)$, where $P_y = \|\boldsymbol{A}\boldsymbol{x}^\star\|^2/m$ is the average power of the noise-free data and P_n is the variance of the noise process $\boldsymbol{n}$.

A. Dictionary and Simulated Data

The impulse response $\boldsymbol{h}$ is a Gaussian signal of standard deviation σ, sampled on a regular grid at integer locations. It is approximated by a finite impulse response of length 6σ by thresholding the smallest values, allowing for fast implementation even for large size problems (see Section IV-C). The deconvolution problem leads to a Toeplitz matrix $\boldsymbol{A}$ whose columns are obtained by shifting the signal $\boldsymbol{h}$. The dimension of $\boldsymbol{A}$ is chosen to have any Gaussian feature resulting from the convolution $\boldsymbol{h} * \boldsymbol{x}^\star$ belonging to the observation window $\{1,\ldots,m\}$. This implies that $\boldsymbol{A}$ is slightly undercomplete $(m > n)$.

B. Separation of Two Close Gaussian Features

We first analyze the ability of SBR to separate two Gaussian features ($\|\boldsymbol{x}^\star\|_0 = 2$) from noise-free data. The centers of both Gaussian features lay at a relative distance d (expressed as a number of samples) and their weights $x_i^\star$ are set to 1. We analyze the SBR outputs for decreasing λ-values by computing their cardinality and testing whether they coincide with the true support $\mathcal{S}(\boldsymbol{x}^\star)$. Table II shows the results obtained for a problem of size 300×270 ($\sigma = 5$) with distances equal to $d = 20$, 13, and 6 samples. It is noticeable that the exact recovery always occurs provided that λ is sufficiently small. This result remains true even for smaller distances (from $d = 2$). When the Gaussian features strongly overlap, i.e., for $d \leqslant 13$, the size of the output support first increases while λ decreases, and then removals start to occur, enabling the exact recovery for lower λ's.

[1] Matlab codes provided by the authors can be downloaded at http://ieeexplore.org. In our Matlab implementation, the insertion and removal tests are done in parallel.

C. Behavior of SBR for Noisy Data

We consider a more realistic simulation in which the data are of larger size ($m = 3000$ samples) and noisy. The impulse response $\boldsymbol{h}$ is of size 301 ($\sigma = 50$) yielding a matrix $\boldsymbol{A}$ of size 3000×2700, and the SNR is set to 20 dB. Fig. 1(a) displays the generated data. The unknown sparse signal $\boldsymbol{x}^\star$ is composed of 17 spikes that are uniformly located in $\{1, \ldots, n\}$. The nonzero amplitudes $x_i^\star$ are drawn according to an i.i.d. Laplacian distribution. Let us remark that the limit BG model is not a proper probabilistic model so that one cannot use it to design simulated data. We choose a Laplacian distribution since the nonzero amplitudes are more heterogeneous than with a Gaussian distribution with finite variance.

In Fig. 1(b)–(d), we display the SBR results for three λ-values. For large λ's, only the main Gaussian features are found. When λ decreases, the smaller features are being recovered together with spurious features. Removals occur for $\lambda \leqslant 0.8$ yielding approximations that are more accurate than those obtained with OLS and for the same cardinality (the residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ is lower) while when $\lambda > 0.8$, the SBR output coincides with the OLS solution of same cardinality. Note that the theoretical value of λ obtained from (3) is equal to 0.3 yielding a support of cardinality 18. The residual is slightly lower than that obtained with $\lambda = 0.5$. The exact support of $\boldsymbol{x}^\star$ is never found because the data are noisy and the neighboring columns of $\boldsymbol{A}$ are highly correlated. In such difficult case, one needs to perform a wider exploration of the discrete set $\{0, 1\}^n$ by introducing moves that are more complex than single replacements. Such extensions were already proposed in the case of SMLR. One can for instance shift an existing spike x_i forwards of backwards [37] or update a block of neighboring amplitudes jointly (e.g., x_i and x_{i+1}) [38]. Various search strategies are also reported in [18, Ch. 3].

D. Comparison of SBR With Other Sparse Algorithms

We compared SBR with classical and recent sparse algorithms: OMP, OLS, CoSaMP [8], subspace pursuit [9], iterative hard thresholding (IHT) [10], [11], ℓ_1 regression [3], and iterative reweighted ℓ_1 (IRℓ_1) [5], [40]. A general trend is that thresholding algorithms perform poorly when the dictionary columns are strongly correlated. CoSaMP and subspace pursuit yield the worst results: they stop after a very few iterations as the squared error increases from one iteration to the next. On the contrary, IHT guarantees that the squared error decreases but the convergence is very slow and the results remain poor in comparison with SBR. In the simulation of Fig. 1(c), SBR performs 12 iterations (only insertions) leading to a support of cardinality 12. Meanwhile, the number of iterations of IHT before convergence is huge: both versions of IHT presented in [10] require at least 10 000 iterations to converge, leading to an overall computation time (22 and 384 s) that is much larger than the SBR computation time (3 s).

Fig. 2 is a synthetic view of the performance of SBR, OLS, OMP, ℓ_1 regression, and IRℓ_1 for a given sparsity level λ. The computation time and the value of $\mathcal{J}(\boldsymbol{x}; \lambda)$ are shown on the horizontal and vertical axes, respectively. This enables us to define several categories of algorithms depending on their locations on the 2-D plane: the outputs of fast algorithms (OMP and ℓ_1) lay in the upper left region whereas slower but more efficient algorithms (OLS, SBR, and IRℓ_1) yield points laying in the lower right region. We chose not to represent the outputs of thresholding algorithms since they yield poorer performance, i.e., points located either in the upper right (IHT) or upper left (CoSaMP, subspace pursuit) regions. In details, we observed that ℓ_1 regression tends to overestimate the support cardinality and to place several spikes at very close locations. We used Donoho's homotopy implementation [3], [39] and found that it requires many iterations: homotopy runs during 200 iterations before reaching a support of cardinality 18 when processing the data of Fig. 1 (we recall that homotopy starts from the empty set and performs a single support replacement per iteration). The performance of ℓ_1 regression fluctuates around that of OMP depending on the trials and the sparsity level. Regarding IRℓ_1, we used the Adaptive LASSO implementation from Zou [40] since it is dedicated to the minimization of $\mathcal{J}(\boldsymbol{x}; \lambda)$. We stopped the algorithm when two successive ℓ_1 iterates share the same support. For the simulation of Fig. 1, IRℓ_1 and SBR yield comparable results in that one algorithm does not outperform the other for all λ values, but IRℓ_1 generally performs slightly better (Fig. 2). We designed other simulations in which the nonzero weights $x_i^\star$ are spread over a wider interval. In this case, SBR most often yields the best approximations.

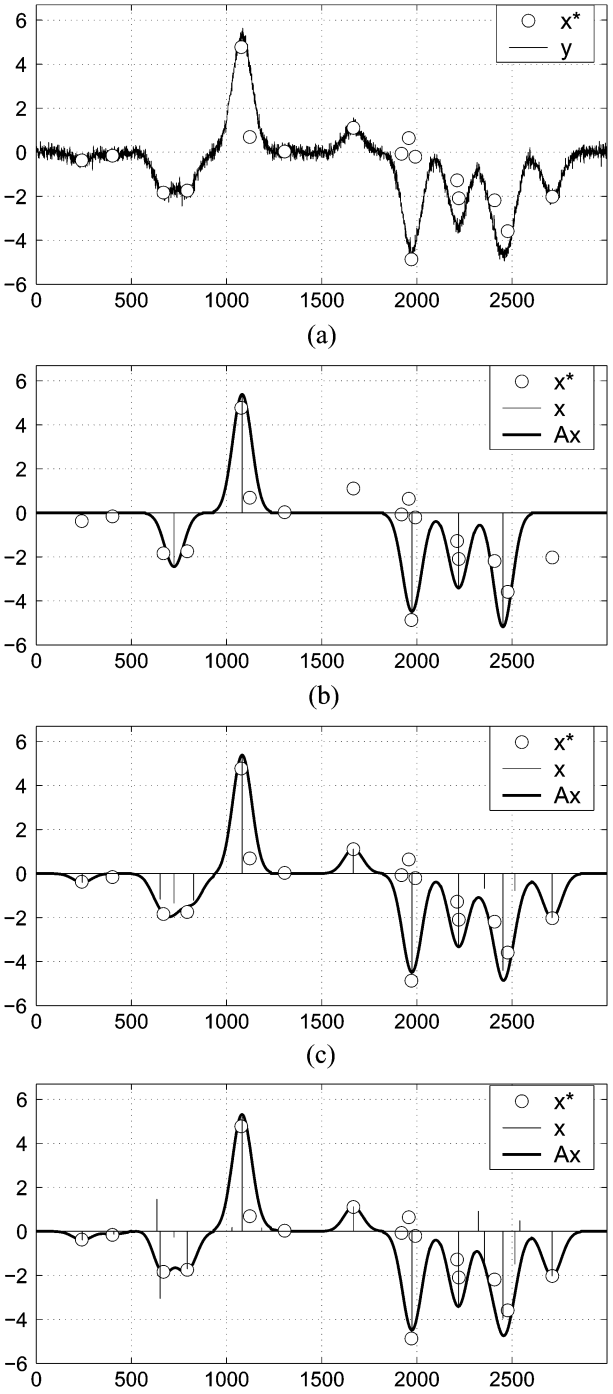
Fig. 2 is representative of the empirical results obtained while performing many trials. Obviously, the figure may significantly change depending on several factors among which the λ-value and the tuning parameters of IRℓ_1. The goal is definitely not to conclude that an algorithm *always* outperforms the others but rather to sketch a classification of groups of algorithms according to the tradeoff between accuracy and computation time.

VI. Joint Detection of Discontinuities at Different Orders in a Signal

We now consider another challenging problem: the joint detection of discontinuities at different orders in a signal [41], [42]. We process both simulated and real data and compare the performance of SBR with respect to OMP, Bayesian OMP (BOMP) which is an OMP based forward–backward algorithm [23], OLS, ℓ_1 regression [3], and IRℓ_1 [5], [7], [40]. First, we formulate the detection of discontinuities at a single order as a spline approximation problem. Then, we take advantage of this formulation to introduce the joint detection problem.

A. Approximation of a Spline of Degree P

Following [41], we introduce the dictionary $\boldsymbol{A}^p$ of size $m \times (m-p)$ formed of shifted versions of the one-sided power function $k \mapsto [\max(k, 0)]^p$ for all possible shifts (see Fig. 3) and we address the sparse approximation of $\boldsymbol{y}$ by the piecewise polynomial $\boldsymbol{A}^p\boldsymbol{x}^p$ (actually, we impose as initial condition that the spline function is equal to 0 for $k \leqslant 0$). It consists in the detection of the discontinuity locations (also referred to as knots in the spline approximation literature) and the estimation of their amplitudes: x_i^p codes for the amplitude of a jump at location i ($p = 0$), the change of slope at location i ($p = 1$), etc. Here, the notion of sparsity is related to the number of discontinuity locations.



(a)

(b)

(c)

(d)

Fig. 1. Gaussian deconvolution results. Problem of size 3000 × 2700 ($\sigma = 50$). (a) Generated data, with 17 Gaussian features and with $\mathrm{SNR} = 20$ dB. The exact locations $\boldsymbol{x}^\star$ are labeled o. (b), (c), (d) SBR outputs and data approximations with empirical settings of λ: $\lambda = 500, 10,$ and 0.5, respectively. The estimated amplitudes $\boldsymbol{x}$ are shown with vertical spikes. The SBR outputs (supports) are of size 5, 12, and 18, respectively. The computation time always remains below 3 s (Matlab implementation).

B. Piecewise Polynomial Approximation

We formulate the joint detection of discontinuities of orders $p = 0, \ldots, P$ by appending the elementary dictionaries $\boldsymbol{A}^p$ in a global dictionary $\boldsymbol{A} = [\boldsymbol{A}^0, \ldots, \boldsymbol{A}^P]$. The product $\boldsymbol{Ax}$ yields a sum of piecewise polynomials of degree lower than P with a limited number of pieces. The dictionary $\boldsymbol{A}$ is overcomplete since it is of size $m \times s$, with $s = (P+1)(m - P/2) > m$ for $P \geqslant 1$. Moreover, any column $\boldsymbol{a}_i^p$ of $\boldsymbol{A}^p$ overlaps *all* other columns $\boldsymbol{a}_j^q$ because their respective supports are the intervals $\{i, \ldots, m\}$ and $\{j, \ldots, m\}$. The discontinuity detection problem is difficult as most algorithms are very likely to position wrong discontinuities in their first iterations. For example, when approximating a signal with two discontinuities at distinct locations i and j, greedy algorithms start to position a first (wrong) discontinuity in between i and j, and forward greedy algorithms cannot remove it.

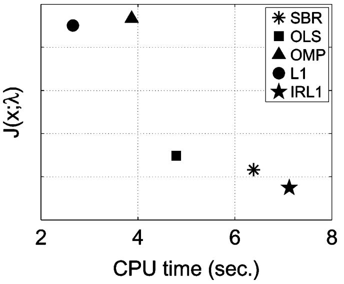


Fig. 2. Comparison of sparse algorithms in terms of tradeoff between accuracy ($\mathcal{J}(\boldsymbol{x};\lambda)$) and CPU time for the deconvolution problem of Fig. 1. SBR($\lambda = 0.5$) is run first yielding a support of cardinality $k_{\mathrm{sbr}} = 18$. Then, we run OLS(k_{sbr}), OMP(k_{sbr}), homotopy for ℓ_1 regression [39], and IR$\ell_1(\lambda)$ [40]. The ℓ_1 result is the homotopy iterate of cardinality k_{sbr} yielding the least value of $\mathcal{J}(\boldsymbol{x};\lambda)$.

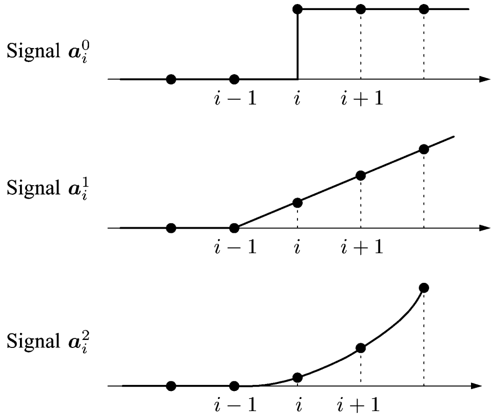
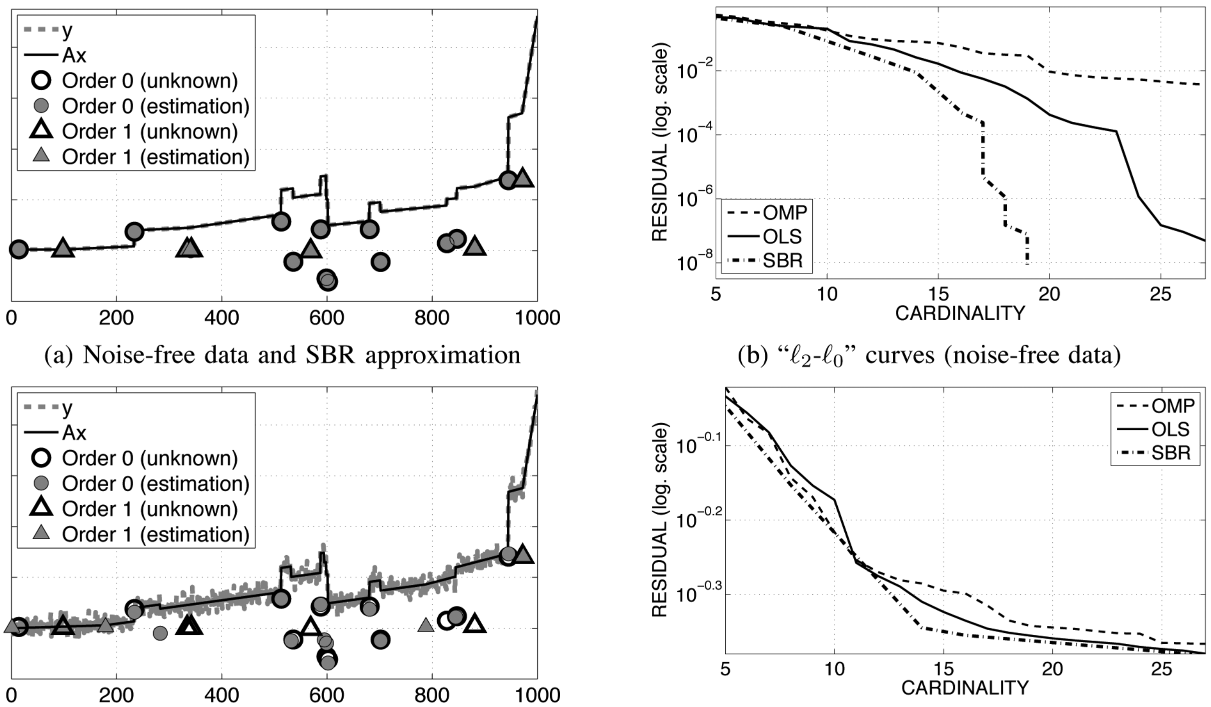

Signal $\boldsymbol{a}_i^0$
Signal $\boldsymbol{a}_i^1$
Signal $\boldsymbol{a}_i^2$
$i-1$
i
$i+1$

Fig. 3. Signals $\boldsymbol{a}_i^p$ related to the pth order discontinuities at location i. $\boldsymbol{a}_i^0$ is the Heaviside step function, $\boldsymbol{a}_i^1$ is the ramp function, and $\boldsymbol{a}_i^2$ is the one-sided quadratic function. Each signal is equal to 1 at location i and its support is equal to $\{i, \ldots, m\}$.

C. Adaptation of SBR

The above defined dictionary does not satisfy the unique representation property. Indeed, it is easy to check that the difference between two discrete ramps at locations i and $i+1$ yields the discrete Heaviside function at location i: $\boldsymbol{a}_i^1 - \boldsymbol{a}_{i+1}^1 = \boldsymbol{a}_i^0$. We thus need to slightly modify SBR in order to ensure that only full rank matrices $\boldsymbol{A}_Q$ are explored. The modification is based on the following proposition which gives a sufficient condition for full rankness of $\boldsymbol{A}_Q$.

Proposition 2: Let n_i denote the number of columns $\boldsymbol{a}_i^p$, $p \in \{0, \ldots, P\}$ which are active for sample i. Let us define the binary condition $\mathcal{C}(i)$:



(a) Noise-free data and SBR approximation

(b) "ℓ_2-ℓ_0" curves (noise-free data)

(c) Noisy data (SNR = 20 dB) and SBR approximation

(d) "ℓ_2-ℓ_0" curves (noisy data)

Fig. 4. Joint detection of discontinuities of orders 0 and 1. The dictionary is of size 1000 × 1999 and the data signal includes 18 discontinuities. The true and estimated discontinuity locations are represented with unfilled black and filled gray labels. The shape of the labels (circular or triangular) indicates the discontinuity order. The dashed gray and solid black curves represent the data signal $\boldsymbol{y}$ and its approximation $\boldsymbol{Ax}$ for the least λ-value. (a) Approximation from noise-free data. The recovery is exact. (b) "ℓ_2-ℓ_0" curves showing the squared residual versus the cardinality for the SBR, OLS, and OMP solutions. (c), (d) Similar results for noisy data ($\mathrm{SNR}\ = 20$ dB).

- if $n_i = 0$, $\mathcal{C}(i) \triangleq 1$;
- if $n_i \geqslant 1, \mathcal{C}(i) \triangleq \{n_{i+j} = 0, j = 1, \ldots, n_i - 1\}$.

If $\mathcal{Q}$ satisfies $\forall i$, $\mathcal{C}(i) = 1$, then $\boldsymbol{A}_{\mathcal{Q}}$ is full rank.

Proposition 2 is proved in Appendix D. Basically, it states that we can allow several discontinuities to be active at the same location i, but then, the next samples $i+1, \ldots, i+n_i-1$ must not host any discontinuity. This condition ensures that there are at most n_i discontinuities in the interval $\{i, \ldots, i+n_i-1\}$ of length n_i. The SBR adaptation consists in testing an insertion only when the new support $\mathcal{Q}' = \mathcal{Q} \cup \{(i,p)\}$ satisfies the above condition.

D. Numerical Simulations

We first set $P = 1$ leading to the piecewise affine approximation problem. The noise-free data $\boldsymbol{y} = \boldsymbol{Ax}^\star$ of Fig. 4(a) are of size $m = 1000$ with $\|\boldsymbol{x}^\star\|_0 = 18$ discontinuities. According to Remark 2, we compute the value $\lambda_{\max}$ above which the SBR output is the empty set, and we run SBR with $\lambda_j = \lambda_{\max} 10^{-j/2}$ for $j = 0, \ldots, 20$. For the least λ-value, SBR yields an exact recovery [see Fig. 4(a)]. For comparison purpose, we also run 27 iterations of OMP and OLS. The "ℓ_2-ℓ_0" curves represented on Fig. 4(b) express the squared residual $\|\boldsymbol{y} - \boldsymbol{Ax}\|^2$ versus the cardinality $\|\boldsymbol{x}\|_0$ for each algorithm (we plot the first 27 iterates of OMP and OLS and for all j, we plot the output of $\mathrm{SBR}(\lambda_j)$ after full convergence of SBR). Whatever the cardinality, SBR yields the least residual. For noisy data, the "ℓ_2-ℓ_0" curve corresponding to SBR still lays below the OMP and OLS curves for most cardinalities. In the next paragraph, we also consider the Bayesian OMP, ℓ_1 regression, and IRℓ_1 algorithms for further comparisons.

E. AFM Data Processing

In atomic force microscopy (AFM), a force curve measures the interatomic forces exerting between a probe associated to a cantilever and a nano-object. Specifically, the recorded signal $z \mapsto y(z)$ shows the force evolution versus the probe-sample distance z, expressed in nanometers. Researching discontinuities (location, order, and amplitude) in a force curve is a challenging task because they are used to provide a precise characterization of the physico-chemical properties of the nano-object (topography, energy of adhesion, etc.) [43].

The data displayed on Fig. 5(a) are related to a bacterial cell *Shewanella putrefaciens* laying in aqueous solution, interacting with the tip of the AFM probe [44]. A retraction force curve is recorded by positioning the tip in contact with the bacterial cell, and then gradually retracting the tip from the sample until it loses contact. In the retraction curve shown on Fig. 5(a), three regions of interest can be distinguished from right to left. The linear region on the right characterizes the rigid contact between the probe and the sample. It describes the mechanical interactions of the cantilever and the sample. The rigid contact is maintained until $z \approx -2840$ nm. The interactions occurring in the interval $z \in [-3050, -2840]$ nm are adhesion forces during the

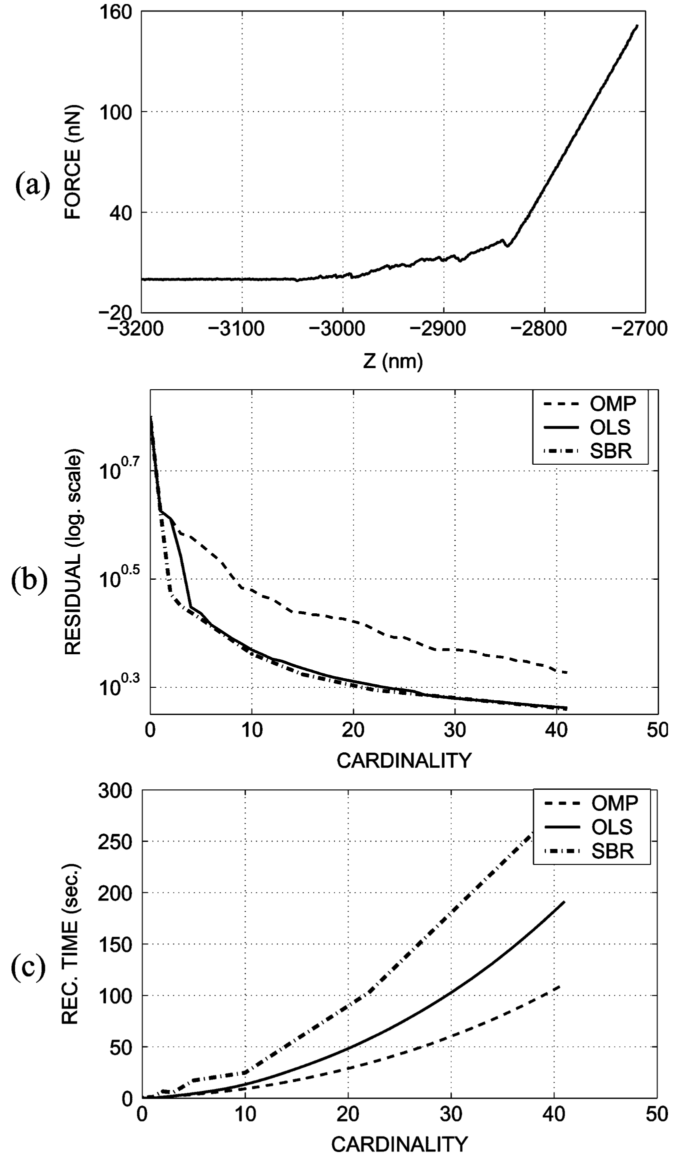


Fig. 5. Joint detection of discontinuities of orders 0, 1, and 2 (problem of size 2167 × 6498). (a) Experimental AFM data showing the force evolution versus the probe-sample distance z. (b) Squared residual versus cardinality for the SBR, OLS, and OMP solutions. (c) Time of reconstruction versus cardinality.

tip retraction. In the flat part on the left, no interaction occurs as the cantilever has lost contact with the sample.

We search for the discontinuities of orders 0, 1, and 2. Similar to the processing of simulated data, we run SBR with 14 λ-values and we run OLS and OMP until iteration 41. For each algorithm, we plot the "ℓ_2-ℓ_0" curve and the curve displaying the time of reconstruction versus the cardinality [Fig. 5(b) and (c)]. These figures show that the performance of SBR is at least equal and sometimes better than that of OLS. Both algorithms yield results that are far more accurate than OMP at the price of a larger computation time.

Fig. 6 displays the approximations yielded by the three algorithms together with the BOMP, ℓ_1, and IRℓ_1 approximations. For the largest value λ_1, SBR runs during six iterations (four insertions and two removals) yielding a support of cardinality 2. SBR performs better than other algorithms [Fig. 6(a)–(f)]. Although IRℓ_1 yields the most accurate approximation, it relies on 4 dictionary columns leading to a larger value of $\mathcal{J}(\boldsymbol{x};\lambda_1)$. We observed the same behavior for the lowest value λ_2 [Fig. 6(g)–(l)]. Again, SBR yields the least value of $\mathcal{J}(\boldsymbol{x};\lambda_2)$ among all algorithms. Moreover, SBR provides a very precise localization of both first order discontinuities [Fig. 6(a)] which are crucial information for the physical interpretation of the data. On the contrary, all other algorithms fail for the highest sparsity level, and some do not even succeed for the lowest. Specifically, OLS accurately locates both first order discontinuities when five iterations have been performed (the desired discontinuities are the first and the last ones among the five) while OMP fails even after five iterations. LASSO and BOMP yield very poor approximations for the highest sparsity level and approximations with many dictionary columns for the lowest sparsity level. In terms of value of the cost function $\mathcal{J}(\boldsymbol{x};\lambda)$, BOMP and LASSO fluctuate around OMP but they are far outperformed by OLS, SBR, and IRℓ_1.

VII. Conclusion

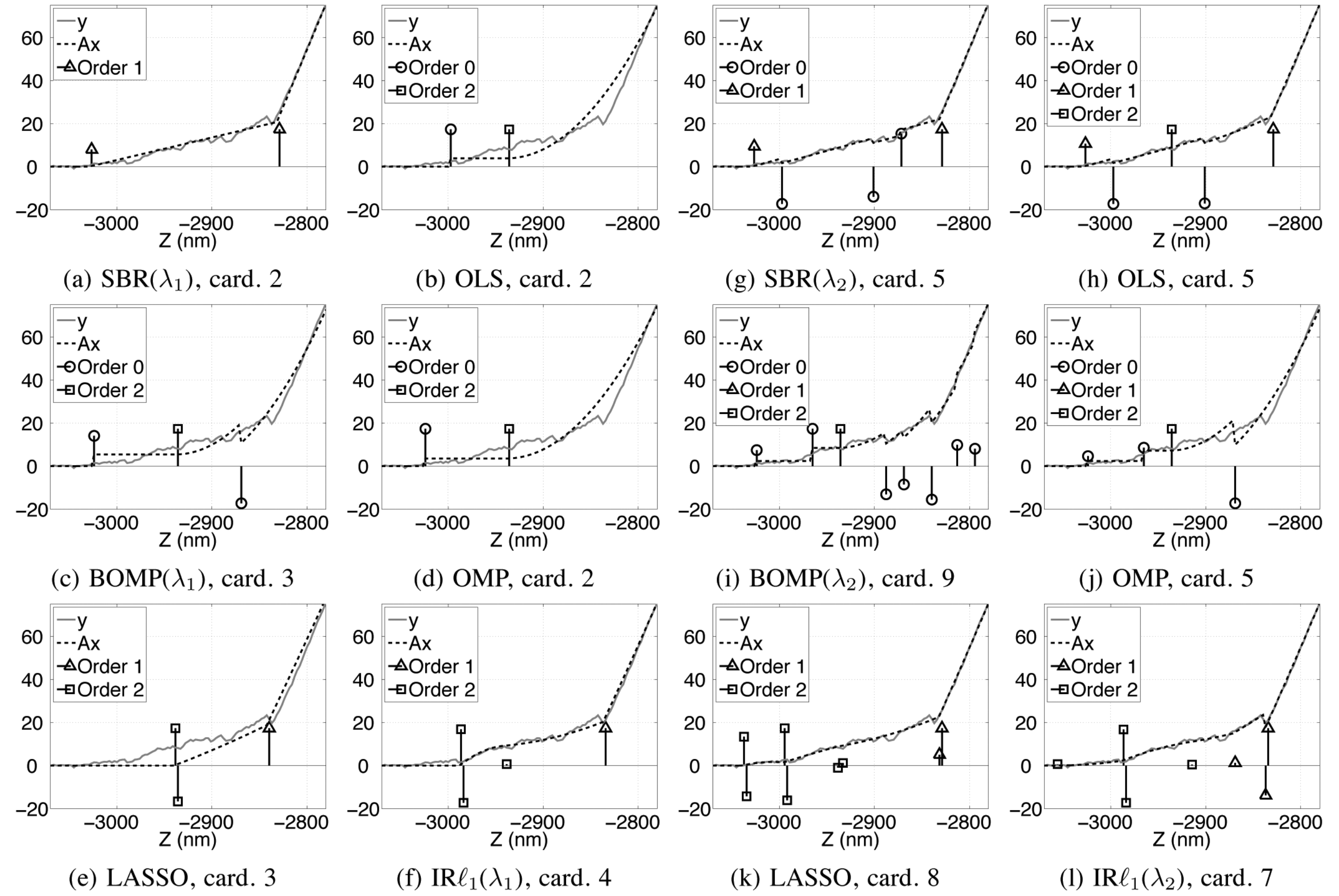
A. Discussion

We performed comparisons for two problems involving highly correlated dictionary columns. SBR is at least as accurate as OLS and sometimes more accurate, with a slightly larger cost of computation. We also considered sparse algorithms that are slower than OLS. SBR was found to be very competitive in terms of tradeoff between accuracy and computation time. Although OLS based forward–backward algorithms yield a relatively large computational cost per iteration, we have noticed that for correlated dictionaries, the number of SBR iterations (i.e., of elementary modifications of the support) is much lower than the number of support modifications performed by several other algorithms. Typically, IHT and IRℓ_1 can often be more expensive than SBR. Additionally, SBR terminates within a finite number of iterations, thus it does not require to tune any empirical stopping parameter. The limitation of SBR in terms of speed arises when the dictionary $\boldsymbol{A}$ is unstructured and the size of $\boldsymbol{A}$ is too large to store $\boldsymbol{A}^t\boldsymbol{A}$. The inner products $\boldsymbol{a}_i^t\boldsymbol{a}_j$ must then be recomputed for each iteration, which is relatively burdensome.

In the recent literature, it is often acknowledged that the cost function $\mathcal{J}(\boldsymbol{x};\lambda)$ has a large number of local minimizers therefore discouraging its direct optimization [5], [7]. Many authors thus choose to minimize an approximate cost function in which the ℓ_0 norm $|x_i|_0$ is replaced with a nonconvex continuous function $\varphi(x_i)$. However, when the range of values of the (expected) nonzero amplitudes $x_i \neq 0$ is wide, it is difficult to find a good approximation $\varphi(x_i)$ of $|x_i|_0$ for all x_i. Selecting an appropriate φ function generally relies on the introduction of a degree of freedom whose tuning is not obvious [5], [6]. For instance, the IRℓ_1 algorithm can be interpreted as an approximate ℓ_2-ℓ_0 minimization method where the ℓ_0 norm is replaced with $\varphi(x_i;\varepsilon) = \log(|x_i| + \varepsilon)$ [5], [7]. The parameter ε controls the "degree of nonconvexity" of the surrogate function φ.[2]

Although $\mathcal{J}(\boldsymbol{x};\lambda)$ has a large number of local minima, we have found that SBR is often as accurate as algorithms based on

[2] In the comparisons with SBR, we set $\varepsilon = 0$ following [40].



(a) SBR(λ_1), card. 2

(b) OLS, card. 2

(g) SBR(λ_2), card. 5

(h) OLS, card. 5

(c) BOMP(λ_1), card. 3

(d) OMP, card. 2

(i) BOMP(λ_2), card. 9

(j) OMP, card. 5

(e) LASSO, card. 3

(f) IRℓ_1(λ_1), card. 4

(k) LASSO, card. 8

(l) IRℓ_1(λ_2), card. 7

Fig. 6. AFM data processing: joint detection of discontinuities of orders 0, 1, and 2. The estimated discontinuities $\boldsymbol{x}$ are represented with vertical spikes and with a label indicating the discontinuity order. (a) SBR output of cardinality 2: four insertions and two removals have been done ($\lambda_1 = 120$). (b)–(f) OLS and OMP outputs after two iterations, BOMP and IRℓ_1 [40] outputs for $\lambda = \lambda_1$, homotopy iterate (LASSO) leading to the minimal value of $\mathcal{J}(\boldsymbol{x}; \lambda_1)$. (g)–(l) Same simulation with a lower λ-value ($\lambda_2 = 8.5$). The SBR output is of cardinality 5 (seven insertions and two removals).

the nonconvex approximation of $\mathcal{J}$. Moreover, SBR is simple to use. The good behavior of SBR is somehow related to the result of Proposition 1 which states that any SBR iterate is almost surely a local minimizer of $\mathcal{J}$. We conclude that SBR is actually capable to "skip" local minima with a large cost $\mathcal{J}(\boldsymbol{x}; \lambda)$.

B. Perspectives

In the proposed approach, the main difficulty relies in the choice of the λ-value. If a specific cardinality or approximation residual is desired, one can resort to a trial and error procedure in which a number of λ-values are tried until the desired approximation level is found. In [45], we sketched a continuation version in which a series of SBR solutions are computed for decreasing levels of sparsity λ, and the λ-values are recursively computed. This continuation version is showing promising results and will be the subject of a future extended contribution. A similar perspective was actually proposed by Zhang to generalize his FoBa algorithm in a path-following algorithm (see the discussion section in [22]).

Another important perspective is to investigate whether SBR can guarantee exact recovery in the noise-free case under some conditions on matrix $\boldsymbol{A}$ and on the unknown sparse signal $\boldsymbol{x}^\star$. According to Remark 1, we will study the behavior of SBR when $\lambda \to 0$. In the simulations done in Sections V and VI, we observed that SBR is able to perform exact recoveries provided that λ is sufficiently small. This promising result is a first step towards a more general theoretical study.

Appendix A
Detailed Development of Limit BG Signal Restoration

Consider the Bernoulli-Gaussian (BG) model $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{r})$ introduced in Section II-B and the joint MAP formulation (3) involving the cost function $\mathcal{L}(\boldsymbol{q}, \boldsymbol{r})$. Given $\boldsymbol{q}$, let us split $\boldsymbol{r}$ into two subvectors $\boldsymbol{u}$ and $\boldsymbol{t}$ indexed by the null and nonnull entries of $\boldsymbol{q}$, respectively. Since $\|\boldsymbol{r}\|^2 = \|\boldsymbol{t}\|^2$ and $\boldsymbol{A}\Delta_{\boldsymbol{q}}\boldsymbol{r} = \boldsymbol{A}_Q\boldsymbol{t}$ do not depend on $\boldsymbol{u}$, we have $\min_{\boldsymbol{u}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{u}) = \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$. Thus, the joint MAP estimation problem reduces to the minimization of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$ w.r.t. $(\boldsymbol{q}, \boldsymbol{t})$. In the limit case $\sigma_x^2 \to \infty$, this problem rereads

$$\min_{\boldsymbol{q}, \boldsymbol{t}} \left\{ \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0}) = \|\boldsymbol{y} - \boldsymbol{A}_Q\boldsymbol{t}\|^2 + \lambda\|\boldsymbol{q}\|_0 \right\}. \tag{9}$$

The equivalence between (9) and (4) directly follows from the change of variable $\boldsymbol{x} = \{\boldsymbol{q}, \boldsymbol{t}\}$ where $\boldsymbol{q}$ and $\boldsymbol{t}$ are the support and nonzero amplitudes of $\boldsymbol{x}$.

Appendix B
Proof of Remark 1

The proof of the result stated in Remark 1 is based on the two following lemmas.

Lemma 1: For $\lambda > 0$, any minimizer of $\mathcal{J}(\boldsymbol{x};\lambda)$ takes the form $\boldsymbol{x}_Q$ with $\|x_Q\|_0 = \mathrm{Card}[\mathcal{Q}] \leqslant \min(m,n)$.

Proof of Lemma: According to the URP assumption, any $\min(m,n)$ columns of $\boldsymbol{A}$ yield an unconstrained minimizer of $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. Let $\boldsymbol{x}_{\mathrm{LS}}$ be such minimizer, with $\|\boldsymbol{x}_{\mathrm{LS}}\|_0 \leqslant \min(m,n)$, and let $\boldsymbol{u}$ be a minimizer of $\mathcal{J}(\boldsymbol{x};\lambda)$. $\mathcal{J}(\boldsymbol{u};\lambda) \leqslant \mathcal{J}(\boldsymbol{x}_{\mathrm{LS}};\lambda)$ implies that $\|\boldsymbol{u}\|_0 \leqslant \|\boldsymbol{x}_{\mathrm{LS}}\|_0 + (\mathcal{E}(\boldsymbol{x}_{\mathrm{LS}}) - \mathcal{E}(\boldsymbol{u}))/\lambda \leqslant \|\boldsymbol{x}_{\mathrm{LS}}\|_0 \leqslant \min(m,n)$.

We denote by $\mathcal{Q}$ the support of $\boldsymbol{u}$. The related least-square solution $\boldsymbol{x}_Q$ obviously satisfies $\mathcal{E}(\boldsymbol{x}_Q) \leqslant \mathcal{E}(\boldsymbol{u})$ and $\|\boldsymbol{x}_Q\|_0 \leqslant \mathrm{Card}[\mathcal{Q}] = \|\boldsymbol{u}\|_0$, thus $\mathcal{J}(\boldsymbol{x}_Q;\lambda) \leqslant \mathcal{J}(\boldsymbol{u};\lambda)$. Since $\boldsymbol{u}$ is a minimizer of $\mathcal{J}(\boldsymbol{x};\lambda)$, we have $\mathcal{J}(\boldsymbol{x}_Q;\lambda) = \mathcal{J}(\boldsymbol{u};\lambda)$ hence $\mathcal{E}(\boldsymbol{x}_Q) = \mathcal{E}(\boldsymbol{u})$. Because of the URP assumption, the least-squares minimizer over $\mathcal{Q}$ is unique, thus $\boldsymbol{u} = \boldsymbol{x}_Q$. ■

Lemma 2: There exists $\lambda_{\min} > 0$ such that for $0 < \lambda \leqslant \lambda_{\min}$, the minimizers of $\mathcal{J}(\boldsymbol{x};\lambda)$ are unconstrained minimizers of $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$.

Proof of Lemma: When λ tends towards 0, we have for all $\mathcal{Q}$, $\mathcal{J}(\boldsymbol{x}_Q;\lambda) = \mathcal{E}_Q + \lambda\|\boldsymbol{x}_Q\|_0 \to \mathcal{E}_Q$. In particular, $\mathcal{J}(\boldsymbol{x}_{Q_{\mathrm{LS}}};\lambda) \to \mathcal{E}_{Q_{\mathrm{LS}}}$ with $\boldsymbol{x}_{Q_{\mathrm{LS}}}$ an unconstrained minimizer of $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ yielded by a subset $\mathcal{Q}_{\mathrm{LS}}$ of cardinality $\min(m,n)$. Because the number of possible subsets $\mathcal{Q}$ is finite and for all $\mathcal{Q}$, $\mathcal{E}_Q \geqslant \mathcal{E}_{Q_{\mathrm{LS}}}$, there exists $\lambda_{\min} > 0$ such that for $0 < \lambda \leqslant \lambda_{\min}$, the subsets $\mathcal{Q}^\star$ minimizing $\mathcal{J}(\boldsymbol{x}_Q;\lambda)$ satisfy $\mathcal{E}_{Q^\star} = \mathcal{E}_{Q_{\mathrm{LS}}}$. Consequently, the minimizers of $\mathcal{J}(\boldsymbol{x};\lambda)$ are unconstrained least-squares solutions according to Lemma 1. ■

Proof of Remark 1: The proof directly follows from the application of Lemma 2. We denote by $\mathcal{X}_\lambda$ the set of minimizers of $\mathcal{J}(\boldsymbol{x};\lambda)$.

In the undercomplete case, there is a unique unconstrained least-square minimizer $\boldsymbol{x}_{\mathrm{LS}}$. Thus, $\mathcal{X}_\lambda = \{\boldsymbol{x}_{\mathrm{LS}}\}$ for $0 \leqslant \lambda \leqslant \lambda_{\min}$.

In the overcomplete case, we denote by $\mathcal{X}^\star$ the set of sparsest solutions to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. To show that $\mathcal{X}_\lambda = \mathcal{X}^\star$ for $0 < \lambda \leqslant \lambda_{\min}$, we consider $\boldsymbol{x} \in \mathcal{X}^\star$ and $\boldsymbol{x}' \in \mathcal{X}_\lambda$. According to Lemma 2, $\boldsymbol{x}'$ satisfies $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}'$, then $\mathcal{J}(\boldsymbol{x}';\lambda) = \lambda\|\boldsymbol{x}'\|_0$. By definition of $\mathcal{X}^\star$, we have $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ and $\mathcal{J}(\boldsymbol{x};\lambda) = \lambda\|\boldsymbol{x}\|_0 \leqslant \mathcal{J}(\boldsymbol{x}';\lambda)$. Because $\boldsymbol{x}' \in \mathcal{X}_\lambda$ is a minimizer of $\mathcal{J}$, we deduce that $\|\boldsymbol{x}'\|_0 = \|\boldsymbol{x}\|_0$, then $\boldsymbol{x}' \in \mathcal{X}^\star$ and $\boldsymbol{x} \in \mathcal{X}_\lambda$. We have proved that $\mathcal{X}_\lambda = \mathcal{X}^\star$ for $0 < \lambda \leqslant \lambda_{\min}$.

Appendix C
Update of the Cholesky Factorization

At each SBR iteration, n linear systems of the form $\boldsymbol{t}_Q \triangleq \boldsymbol{G}_Q^{-1}\boldsymbol{A}_Q^t\boldsymbol{y}$ must be solved, the corresponding squared errors reading $\mathcal{E}_Q = \|\boldsymbol{y} - \boldsymbol{A}_Q\boldsymbol{t}_Q\|^2 = \|\boldsymbol{y}\|^2 - \boldsymbol{y}^t\boldsymbol{A}_Q\boldsymbol{t}_Q$. Using the Cholesky factorization $\boldsymbol{G}_Q = \boldsymbol{L}_Q\boldsymbol{L}_Q^t$, $\boldsymbol{t}_Q$ rereads $\boldsymbol{t}_Q = \boldsymbol{L}_Q^{-t}\boldsymbol{L}_Q^{-1}\boldsymbol{A}_Q^t\boldsymbol{y}$, thus

$$\mathcal{E}_Q = \|\boldsymbol{y}\|^2 - \left\|\boldsymbol{L}_Q^{-1}\boldsymbol{A}_Q^t\boldsymbol{y}\right\|^2. \tag{10}$$

Insertion of a New Column After the Existing Columns: Including a new column leads to $\boldsymbol{A}_{Q'} = [\boldsymbol{A}_Q, \boldsymbol{a}_i]$. Thus, the new Gram matrix reads as a 2×2 block matrix:

$$\boldsymbol{G}_{Q'} = \begin{bmatrix} \boldsymbol{G}_Q & \boldsymbol{A}_Q^t\boldsymbol{a}_i \\ (\boldsymbol{A}_Q^t\boldsymbol{a}_i)^t & \|\boldsymbol{a}_i\|^2 \end{bmatrix}$$

and the Cholesky factor of $\boldsymbol{G}_{Q'}$ can be straightforwardly updated:

$$\boldsymbol{L}_{Q'} = \begin{bmatrix} \boldsymbol{L}_Q & \boldsymbol{0} \\ \boldsymbol{l}_{Q,i}^t & \sqrt{\|\boldsymbol{a}_i\|^2 - \|\boldsymbol{l}_{Q,i}\|^2} \end{bmatrix} \tag{11}$$

with $\boldsymbol{l}_{Q,i} = \boldsymbol{L}_Q^{-1}\boldsymbol{A}_Q^t\boldsymbol{a}_i$. The update (8) of $\mathcal{J}_Q(\lambda) = \mathcal{E}_Q + \lambda\mathrm{Card}[\mathcal{Q}]$ directly follows from (10) and (11).

Removal of an Arbitrary Column: When removing a column $\boldsymbol{a}_i$, updating $\boldsymbol{L}_Q$ remains possible although more complex. This idea was developed by Ge *et al.* [46] who update the Cholesky factorization of matrix $\boldsymbol{G}_Q^{-1}$. We adapt it to the direct (simpler) factorization of $\boldsymbol{G}_Q$. Let I be the position of $\boldsymbol{a}_i$ in $\boldsymbol{A}_Q$ (with $1 \leqslant I \leqslant \mathrm{Card}[\mathcal{Q}]$). $\boldsymbol{L}_Q$ can be written in a block matrix form

$$\boldsymbol{L}_Q = \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{b}^t & d & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{e} & \boldsymbol{F} \end{bmatrix} \tag{12}$$

where the lowercase characters refer to the scalar (d) and vector quantities $(\boldsymbol{b}, \boldsymbol{e})$ appearing in the Ith row and in the Ith column. The computation of $\boldsymbol{G}_Q = \boldsymbol{L}_Q\boldsymbol{L}_Q^t$ and the removal of the Ith row and the Ith column in $\boldsymbol{G}_Q$ lead to

$$\boldsymbol{G}_{Q'} = \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{F} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^t & \boldsymbol{C}^t \\ \boldsymbol{0} & \boldsymbol{F}^t \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e} \end{bmatrix} [\boldsymbol{0} \quad \boldsymbol{e}^t].$$

By identification with $\boldsymbol{G}_{Q'} = \boldsymbol{L}_{Q'}\boldsymbol{L}_{Q'}^t$ and because the Cholesky factorization is unique, $\boldsymbol{L}_{Q'}$ necessarily reads

$$\boldsymbol{L}_{Q'} = \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{X} \end{bmatrix} \tag{13}$$

where $\boldsymbol{X}$ is a lower triangular matrix satisfying $\boldsymbol{X}\boldsymbol{X}^t = \boldsymbol{F}\boldsymbol{F}^t + \boldsymbol{e}\boldsymbol{e}^t$. The problem of computing $\boldsymbol{X}$ from $\boldsymbol{F}$ and $\boldsymbol{e}$ is classical; it is known as a positive rank 1 Cholesky update and there exists a stable algorithm in $\mathcal{O}(f^2)$ operations, where $f = \mathrm{Card}[\mathcal{Q}] - I$ is the size of $\boldsymbol{F}$ [36].

Appendix D
Proof of Proposition 2

Let us first introduce some notations specific to the piecewise polynomial dictionary problem. Consider a subset $\mathcal{Q}$ of columns $\boldsymbol{a}_i^p$ and let $i^- = \min\{i \,|\, n_i > 0\}$ denote the lowest location of an active entry (we recall that n_i denotes the number of active columns for sample i). Up to a reordering of the columns of $\boldsymbol{A}_Q$, $\boldsymbol{A}_Q$ rereads $\boldsymbol{A}_Q = [\boldsymbol{A}_{i^-}, \tilde{\boldsymbol{A}}_{i^-}]$ where $\boldsymbol{A}_{i^-}$ gathers the n_{i^-} active columns $\boldsymbol{a}_i^p$ such that $i = i^-$ and $\tilde{\boldsymbol{A}}_{i^-}$ gathers the remaining active columns (with $i > i^-$). The following lemma is a key element to prove Proposition 2.

Lemma 3: Assume that $\mathcal{Q}$ satisfies the condition of Proposition 2. If $\tilde{\boldsymbol{A}}_{i^-}$ is full rank, then $\boldsymbol{A}_Q$ is full rank.

Proof: Let $I = n_{i^-}$ denote the number of discontinuities at location i^- and let $0 \leqslant p_1 < p_2 < \cdots < p_I$

denote their orders, sorted in the ascending order. Suppose that there exist two families of scalars $\{\mu_{i^-}^{p_1}, \ldots, \mu_{i^-}^{p_I}\}$ and $\{\mu_i^p \mid i \neq i^- \text{ and } i \text{ is active at order } p\}$ such that

$$\sum_{j=1}^{I} \mu_{i^-}^{p_j} \, \boldsymbol{a}_{i^-}^{p_j} + \sum_{i \neq i^-} \sum_{p} \mu_i^p \, \boldsymbol{a}_i^p = \mathbf{0}. \tag{14}$$

Let us show that all μ-values are then equal to 0.

Rewriting the first I nonzero equations in this system and because $\mathcal{Q}$ satisfies the condition of Proposition 2, we have, for all $k \in \{i^-, \ldots, i^- + I - 1\}$, $\sum_{j=1}^{I} \mu_{i^-}^{p_j} (k + i^- - 1)^{p_j} = 0$. Hence, the polynomial $F(X) = \sum_{j=1}^{I} \mu_{i^-}^{p_j} X^{p_j}$ has I positive roots. Because any nonzero polynomial formed of I monomials of different degrees has at most $I - 1$ positive roots [47, p. 76], F is the zero polynomial, thus all scalars $\mu_{i^-}^{p_j}$ are 0. We deduce from (14) and from the full rankness of $\tilde{\boldsymbol{A}}_{i^-}$ that $\mu_i^p = 0$ for all (i, p).

We have shown that the column vectors of $\boldsymbol{A}_{\mathcal{Q}}$ are linearly independent, i.e., that $\boldsymbol{A}_{\mathcal{Q}}$ is full rank. ■

The proof of Proposition 2 directly results from the recursive application of Lemma 3. Starting from the empty set, all the indices, sorted by decreasing order, are successively included.

ACKNOWLEDGMENT

The authors would like to thank Dr. G. Francius from LCPME (UMR CNRS 7564, Nancy, France) for providing them with real AFM data.

REFERENCES

[1] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[2] 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.

[3] 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.

[4] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Trans. Signal Process*, vol. 51, no. 3, pp. 760–770, Mar. 2003.

[5] 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.

[6] 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.

[7] D. P. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process. (Special Issue on Compressive Sensing)*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[8] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[9] 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.

[10] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 629–654, Dec. 2008.

[11] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, Apr. 2010.

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

[13] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, Nov. 1993, vol. 1, pp. 40–44.

[14] C. Couvreur and Y. Bresler, "On the optimallity of the backward greedy algorithm for the subset selection problem," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 3, pp. 797–808, Feb. 2000.

[15] M. A. Efroymson, "Multiple regression analysis," in *Mathematical Methods for Digital Computers*, A. Ralston and H. S. Wilf, Eds. New York: Wiley, 1960, vol. 1, pp. 191–203.

[16] K. N. Berk, "Forward and backward stepping in variable selection," *J. Stat. Comput. Simulat.*, vol. 10, no. 3–4, pp. 177–185, Apr. 1980.

[17] P. M. T. Broersen, "Subset regression with stepwise directed search," *J. R. Stat. Soc. C*, vol. 35, no. 2, pp. 168–177, 1986.

[18] A. J. Miller, *Subset Selection in Regression*, 2nd ed. London, U.K.: Chapman & Hall, Apr. 2002.

[19] 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.

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

[21] D. Haugland, *A Bidirectional Greedy Heuristic for the Subspace Selection Problem, Lect. Notes Comput. Sci.* Berlin, Germany: Springer Verlag, 2007, vol. 4638, pp. 162–176.

[22] T. Zhang, "Adaptive forward–backward greedy algorithm for learning sparse representations," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, Jul. 2011.

[23] C. Herzet and A. Drémeau, "Bayesian pursuit algorithms," in *Proc. Eur. Signal Process. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 1474–1478.

[24] 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, May 1982.

[25] J. M. Mendel, *Optimal Seismic Deconvolution*. New York: Academic, 1983.

[26] 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.

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

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

[29] Q. Cheng, R. Chen, and T.-H. Li, "Simultaneous wavelet estimation and deconvolution of reflection seismic signals," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, pp. 377–384, Mar. 1996.

[30] E. I. George and D. P. Foster, "Calibration and empirical Bayes variable selection," *Biometrika*, vol. 87, no. 4, pp. 731–747, 2000.

[31] H. Chipman, E. I. George, and R. E. McCulloch, "The practical implementation of Bayesian model selection," *IMS Lecture Notes—Monograph Series*, vol. 38, pp. 65–134, 2001.

[32] J. Nocedal and S. J. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag, 1999.

[33] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Process.*, vol. 43, no. 7, pp. 1713–1715, Jul. 1995.

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

[35] S. J. Reeves, "An efficient implementation of the backward greedy algorithm for sparse signal reconstruction," *IEEE Signal Process. Lett.*, vol. 6, no. 10, pp. 266–268, Oct. 1999.

[36] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," *Math. Comput.*, vol. 28, no. 126, pp. 505–535, Apr. 1974.

[37] C. Y. Chi and J. M. Mendel, "Improved maximum-likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 30, pp. 429–435, Mar. 1984.

[38] M. Allain and J. Idier, "Efficient binary reconstruction for non-destructive evaluation using gammagraphy," *Inverse Problems*, vol. 23, no. 4, pp. 1371–1393, Aug. 2007.

[39] D. L. Donoho, V. Stodden, and Y. Tsaig, "About SparseLab," Stanford Univ., Stanford, CA, Tech. Rep., Mar. 2007.

[40] H. Zou, "The adaptive Lasso and its oracle properties," *J. Acoust. Soc. Amer.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[41] M. S. Smith and R. Kohn, "Nonparametric regression using Bayesian variable selection," *J. Econometrics*, vol. 75, no. 2, pp. 317–343, Dec. 1996.

[42] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1417–1428, Jun. 2002.

[43] H.-J. Butt, B. Cappella, and M. Kappl, "Force measurements with the atomic force microscope: Technique, interpretation and applications," *Surf. Sci. Rep.*, vol. 59, no. 1–6, pp. 1–152, Oct. 2005.

[44] F. Gaboriaud, B. S. Parcha, M. L. Gee, J. A. Holden, and R. A. Strugnell, "Spatially resolved force spectroscopy of bacterial surfaces using force-volume imaging," *Colloids Surf. B.*, vol. 62, no. 2, pp. 206–213, Apr. 2008.

[45] 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.

[46] D. Ge, J. Idier, and E. Le Carpentier, "Enhanced sampling schemes for MCMC based blind Bernoulli-Gaussian deconvolution," *Signal Process.*, vol. 91, no. 4, pp. 759–772, Apr. 2011.

[47] F. R. Gantmacher and M. G. Krein*, Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, rev. ed. Providence, RI: AMS Chelsea, 2002.

Charles Soussen was born in France in 1972. He received the Diploma degree from the École Nationale Supérieure en Informatique et Mathématiques Appliquées, Grenoble, France, and the Ph.D. degree in physics from the Laboratoire des Signaux et Systèmes, Université de Paris-Sud, Orsay, France, in 1996 and 2000, respectively.

He is currently an Assistant Professor at Nancy-University, France. He has been with the Centre de Recherche en Automatique de Nancy, France, since 2005. His research interests are in inverse problems and sparse approximation.

Jérôme Idier (M'09) was born in France in 1966. He received the Diploma degree in electrical engineering from École Supérieure d'Électricité, Gif-sur-Yvette, France, in 1988 and the Ph.D. degree in physics from University of Paris-Sud, Orsay, France, in 1991.

In 1991, he joined the Centre National de la Recherche Scientifique. He is currently a Senior Researcher at the Institut de Recherche en Communications et Cybernétique in Nantes. His major scientific interests are in probabilistic approaches to inverse problems for signal and image processing.

Dr. Idier is serving as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

David Brie received the Ph.D. degree and the Habilitation à Diriger des Recherches degree, both from the Henri Poincaré University, Nancy, France, in 1992 and 2000, respectively.

He is currently Professor at the Telecommunication and Network Department from the Institut Universitaire de Technologie, Nancy-University, France. Since 1990, he has been with the Centre de Recherche en Automatique de Nancy, France. His research interests mainly concern inverse problems and multidimensional signal processing.

Junbo Duan was born in China in 1981. He received the B.S. degree in information engineering and M.S. degree in communication and information system from Xi'an Jiaotong University, China, in 2004 and 2007, respectively, and the Ph.D. degree in signal processing from Université Henry Poincaré, Nancy, France, in 2010.

He is currently a Postdoctoral Researcher in the Department of Biomedical Engineering and Biostatistics, Tulane University, New Orleans, LA. His major research interests are in probabilistic approaches to inverse problems in bioinformatics.