Bayesian Pursuit Algorithms

Cedric Herzet, Angélique Drémeau

Bayesian Pursuit Algorithms

Cédric Herzet[(1)] and Angélique Drémeau[(2)]

[(1)] INRIA-Rennes, Centre Bretagne Atlantique, Rennes, France, cedric.herzet@inira.fr

[(2)] Institut Télécom, Télécom ParisTech, CNRS-LTCI, Paris, France, angelique.dremeau@telecom-paristech.fr

Abstract

This paper addresses the sparse representation (SR) problem within a general Bayesian framework. We show that the Lagrangian formulation of the standard SR problem, *i.e.*, $\mathbf{x}^\star = \arg\min_{\mathbf{x}}\{\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0\}$, can be regarded as a limit case of a general maximum a posteriori (MAP) problem involving Bernoulli-Gaussian variables. We then propose different tractable implementations of this MAP problem that we refer to as "Bayesian pursuit algorithms". The Bayesian algorithms are shown to have strong connections with several well-known pursuit algorithms of the literature (*e.g.*, MP, OMP, StOMP, CoSaMP, SP) and generalize them in several respects. In particular, *i)* they allow for atom *deselection*; *ii)* they can include any prior information about the probability of occurrence of each atom within the selection process; *iii)* they can encompass the estimation of unkown model parameters into their recursions.

I. Introduction

Sparse representations (SR) aim at describing a signal as the combination of a small number of *atoms*, namely elementary signals, chosen from a given dictionary. More precisely, let $\mathbf{y} \in \mathbb{R}^N$ be an observed signal and $\mathbf{D} \in \mathbb{R}^{N\times M}$ a dictionary of atoms. Then, one standard formulation of the sparse representation problem writes

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{1}$$

where $\|\cdot\|_0$ denotes the l_0 pseudo-norm, which counts the number of non-zero elements in $\mathbf{x}$, and $\lambda > 0$ is a parameter specifying the trade-off between sparsity and distortion.

Sparse representations have been shown to be relevant in many practical situations. A few examples include statistical regression [1], digital communications [2], image processing [3], [4], interpolation/extrapolation [5], signal deconvolution [6], [7], Tomo PIV [8], compressive sampling [9], etc.

Unfortunately, finding the exact solution of (1) is a NP-hard problem [5], *i.e.*, it generally requires a combinatorial search over the entire solution space. For problems of moderate-to-high dimensionality,

combinatorial approaches are intractable and one has therefore to resort to heuristic procedures. In the current literature, three main families of algorithms can roughly be distinguished: the algorithms based on a problem relaxation, the pursuit algorithms, and the Bayesian algorithms.

The *SR algorithms based on a problem relaxation* approximate the non-smooth and non-convex ℓ_0-norm by functions easier to handle. The resulting problem can then be solved by means of standard optimization techniques. Well-known instances of algorithms based on such an approach are Basis Pursuit (BP) [10] and FOCUSS [11] which approximate the ℓ_0-norm by the ℓ_1- and ℓ_p- ($p < 1$) norms, respectively.

The family of *pursuit algorithms* encompasses all the procedures looking for a solution of the sparse representation problem by making a succession of greedy decisions on the support *i.e.*, by iteratively selecting or deselecting atoms from a "local" perspective. A non-exhaustive list of algorithms belonging to this family includes matching pursuit (MP) [12], orthogonal matching pursuit (OMP) [13], stagewise OMP (StOMP) [14], orthogonal least square (OLS) [15], gradient pursuit (GP) [16], iterative hard thresholding (IHT) [17], hard thresholding pursuit (HTP) [18], compressive sampling matching pursuit (CoSaMP) [19] or subspace pursuit (SP) [20]. In this paper, we will more particularly focus on the family of *forward/backward* algorithms, that is procedures which consider both atom selection and deselection during the estimation process.

Finally, *Bayesian algorithms* express the SR problem as the solution of a Bayesian estimation problem. One key ingredient of the Bayesian algorithms is the choice of a proper prior, enforcing sparsity on the sought vector. A popular approach consists in modelling $\mathbf{x}$ as a continuous random variable whose distribution has a sharp peak to zero and heavy tails [21]–[26]. Another approach, recently gaining in popularity, is based on a prior made up of the combination of Bernoulli and Gaussian distributions. This model has been exploited in the following contributions [6], [7], [27]–[34] and will be considered in this paper.

Bayesian approaches have recently gained in popularity because they allow to effectively account for uncertainties on the model parameters or possible connections between the non-zero elements of the sparse vector (*e.g.*, in the case of structured sparsity). On the other hand, pursuit algorithms are usually attractive because of their good compromise between complexity and performance. The work presented in this paper lies at the intersection of the families of Bayesian and pursuit algorithms. The contributions of the paper are threefold. First, we emphasize a connection between the standard problem (1) and a maximum a posteriori (MAP) problem involving Bernoulli-Gaussian (BG) variables. In particular, we show that the set of solutions of the standard problem and the BG MAP problem are the same for certain values of the parameters of the BG model. Second, we propose four different procedures searching for

the solution of the considered MAP estimation problem. Finally, we emphasize the link existing between the proposed procedures and well-known pursuit algorithms of the literature. In particular, MP, OMP, StOMP and SP are shown to correspond to particular cases of the proposed algorithms for some values of the model parameters.

The rest of the paper is organized as follows. In section III, we present the BG probabilistic model considered in this paper and establish a connection between (1) and a MAP problem involving this model. Section IV is devoted to the derivation of the proposed sparse-representation algorithms. In section V, we recast our contributions within the current literature on Bayesian and pursuit algorithms; we emphasize moreover the connection between the proposed algorithms and some well-known pursuit procedures. Finally, in section VI we provide extensive simulation results comparing, according to different figures of merit, the proposed procedures and several algorithms of the state of the art.

II. Notations

The notational conventions adopted in this paper are as follows. The ith element of vector $\mathbf{a}$ is denoted a_i; $\langle \mathbf{a}, \mathbf{b} \rangle \triangleq \mathbf{a}^T \mathbf{b}$ defines the scalar product between vectors $\mathbf{a}$ and $\mathbf{b}$; $\|\mathbf{a}\| \triangleq \langle \mathbf{a}, \mathbf{a} \rangle^{1/2}$ is the ℓ_2-norm of $\mathbf{a}$; $\|\mathbf{a}\|_0$ denotes the number of non-zero elements in $\mathbf{a}$. The Moore-Penrose pseudo-inverse of matrix $\mathbf{A}$ is denoted by $\mathbf{A}^\dagger$ and we use the notation $\mathbf{I}_N$ for the $N \times N$-identity matrix. The minimum of a function $f(\mathbf{a})$ is denoted by $\min_{\mathbf{a}} f(\mathbf{a})$ and the *set* of values at which this minimum is achieved by $\arg\min_{\mathbf{a}} f(\mathbf{a})$. With a slight abuse of notation, we will often use $\mathbf{a}^\star = \arg\min_{\mathbf{a}} f(\mathbf{a})$ to specify that $\mathbf{a}^\star$ belongs to the set of solutions, *i.e.*, $\mathbf{a}^\star \in \arg\min_{\mathbf{a}} f(\mathbf{a})$.

III. A Bayesian formulation of the standard SR problem

In this section, we present the probabilistic model that will be considered throughout this paper and state a result relating the standard formulation of the SR problem (1) to a MAP estimation problem involving this model.

Let $\mathbf{D} \in \mathbb{R}^{N \times M}$ be a dictionary whose columns are normalized to 1. Let moreover $\mathbf{s} \in \{0,1\}^M$ be a vector defining the *support* of the sparse representation, *i.e.*, the subset of columns of $\mathbf{D}$ used to generate $\mathbf{y}$. We adopt the following convention: if $s_i = 1$ (resp. $s_i = 0$), the ith column of $\mathbf{D}$ is (resp. is not) used to form $\mathbf{y}$. Denoting by $\mathbf{d}_i$ the ith column of $\mathbf{D}$, we then consider the following observation model:

$$\mathbf{y} = \sum_{i=1}^{M} s_i \, x_i \, \mathbf{d}_i + \mathbf{w}, \tag{2}$$

where $\mathbf{w}$ is a zero-mean white Gaussian noise with variance σ_w^2. Therefore,

$$p(\mathbf{y}|\mathbf{x},\mathbf{s}) = \mathcal{N}(\mathbf{D_s x_s}, \sigma_w^2 \mathbf{I}_N), \tag{3}$$

where $\mathbf{D_s}$ (resp. $\mathbf{x_s}$) is a matrix (resp. vector) made up of the $\mathbf{d}_i$'s (resp. x_i's) such that $s_i = 1$; $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. We suppose that $\mathbf{x}$ and $\mathbf{s}$ obey the following probabilistic model:

$$p(\mathbf{x}) = \prod_{i=1}^{M} p(x_i), \qquad\qquad p(\mathbf{s}) = \prod_{i=1}^{M} p(s_i), \tag{4}$$

where

$$p(x_i) = \mathcal{N}(0, \sigma_x^2), \qquad p(s_i) = \mathrm{Ber}(p_i), \tag{5}$$

and $\mathrm{Ber}(p_i)$ denotes a Bernoulli distribution of parameter p_i.

It is important to note that (3)-(5) only define a *model* on $\mathbf{y}$ and may not correspond to its actual distribution. Despite this fact, it is worth noticing that the BG model (3)-(5) is well-suited to modelling situations where $\mathbf{y}$ stems from a sparse process. Indeed, if $p_i \ll 1 \ \forall \ i$, only a small number of s_i's will *typically*[1] be non-zero, *i.e.*, the observation vector $\mathbf{y}$ will be generated with high probability from a small subset of the columns of $\mathbf{D}$. In particular, if $p_i = p \ \forall \ i$, typical realizations of $\mathbf{y}$ will involve a combination of pM columns of $\mathbf{D}$.

We emphasize hereafter a connection between the standard problem (1) and a MAP problem involving model (3)-(5):

Theorem 1: *Consider the following MAP estimation problem:*

$$(\hat{\mathbf{x}}, \hat{\mathbf{s}}) = \underset{(\mathbf{x},\mathbf{s})}{\arg\max} \ \log p(\mathbf{y}, \mathbf{x}, \mathbf{s}), \tag{6}$$

where $p(\mathbf{y}, \mathbf{x}, \mathbf{s}) = p(\mathbf{y}|\mathbf{x}, \mathbf{s})\, p(\mathbf{x})\, p(\mathbf{s})$ *is defined by the Bernoulli-Gaussian model* (3)-(5)*.*
If $\|\mathbf{y}\|_2 < \infty$ *and*

[1]In an information-theoretic sense [35], *i.e.*, according to model (3)-(5), a realization of $\mathbf{s}$ with a few non-zero components will be observed with probability almost 1.

$$\begin{aligned} &\sigma_x^2 \to \infty, \\ &p_i = p \quad \forall\ i,\ p \in [0,1], \\ &\lambda = 2\sigma_w^2 \log\left(\frac{1-p}{p}\right), \end{aligned} \tag{7}$$

the BG MAP problem (6) *and the standard SR problem* (1) *lead to the same set of solutions.* □

A proof of this result can be found in Appendix A. The result established in Theorem 1 recasts the standard sparse representation problem (1) into a more general Bayesian framework. In particular, it shows that (1) is equivalent to a MAP estimation problem for particular values of the model parameters. In the general case, the Bayesian formulation (6) allows for more degrees of freedom than (1). For example, any prior information about the amplitude of the non-zero coefficients (σ_x^2) or the atom occurrence (p_i's) can explicitly be taken into account.

Let us mention that a result similar to Theorem 1 was already presented in our conference paper [32] and the parallel work by Soussen *et al.* [7]. The equivalence proposed in this paper is however more general since, unlike these results, it does not require any condition of the type

$$\|\mathbf{D}_{\mathbf{s}}^{\dagger}\mathbf{y}\|_0 = \|\mathbf{s}\|_0 \quad \forall\, \mathbf{s} \in \{0,1\}^M \tag{8}$$

to hold. In particular, Theorem 1 extends the equivalence between (1) and (6) to the important case of noise-free data. Indeed, assume that the observed vector is generated as follows:

$$\mathbf{y} = \mathbf{D}_{\tilde{\mathbf{s}}}\tilde{\mathbf{x}}_{\tilde{\mathbf{s}}}, \tag{9}$$

where $\tilde{\mathbf{s}} \in \{0,1\}^M$, $\|\tilde{\mathbf{s}}\|_0 < N$, and $\tilde{\mathbf{x}} \in \mathbb{R}^M$ are realizations of some arbitrary random variables. Then, any vector $\mathbf{s}$ such that

$$\begin{cases} s_i = 1 & \text{if } \tilde{s}_i = 1, \\ \|\mathbf{s}\|_0 \leq N, \end{cases} \tag{10}$$

violates the equality (8) and the results in [7], [32] do therefore not apply.

IV. Bayesian Pursuit Algorithms

The BG MAP problem (6) does not offer any advantage in terms of complexity with respect to (1), since it is also NP-hard. Hence, the resolution of (6) requires to resort to heuristic (but practical) algorithms. In this section, we propose several greedy procedures searching for a solution of (6) by generating a sequence

Initialization : $\hat{\mathbf{x}}^{(0)} = 0$, $\hat{\mathbf{s}}^{(0)} = 0$, $n = 0$.
Repeat :

1. Update the residual:

$$\mathbf{r}^{(n)} = \mathbf{y} - \mathbf{D}\hat{\mathbf{x}}^{(n)}. \tag{11}$$

2. Evaluate $\tilde{x}_i^{(n+1)}$ and $\tilde{s}_i^{(n+1)}$ $\forall i$:

$$\tilde{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_i^{(n)}\mathbf{d}_i, \mathbf{d}_i \rangle^2 > T_i, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

$$\tilde{x}_i^{(n+1)} = \tilde{s}_j^{(n+1)} \frac{\sigma_x^2}{\sigma_x^2 + \sigma_w^2} \left(\hat{x}_i^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle \right), \tag{13}$$

with

$$T_i \triangleq 2\sigma_w^2 \frac{\sigma_x^2 + \sigma_w^2}{\sigma_x^2} \log\left(\frac{1 - p_i}{p_i} \right). \tag{14}$$

3. Choose the index to be modified:

$$j = \arg\max_i \rho^{(n)}(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}), \tag{15}$$

where

$$\rho^{(n)}(x_i, s_i) = -\left\| \mathbf{r}^{(n)} + (\hat{s}_i^{(n)}\hat{x}_i^{(n)} - s_i x_i)\mathbf{d}_i \right\|^2 - \epsilon\, x_i^2 - \lambda_i s_i, \tag{16}$$

4. Update the support and the coefficients:

$$\hat{s}_i^{(n+1)} = \begin{cases} \tilde{s}_i^{(n+1)} & \text{if } i = j, \\ \hat{s}_i^{(n)} & \text{otherwise,} \end{cases} \tag{17}$$

$$\hat{x}_i^{(n+1)} = \begin{cases} \tilde{x}_i^{(n+1)} & \text{if } i = j, \\ \hat{x}_i^{(n)} & \text{otherwise.} \end{cases} \tag{18}$$

TABLE I
BMP ALGORITHM

of estimates $\{\hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)}\}_{n=0}^{\infty}$. The first two procedures are particular instances of block-coordinate ascent algorithms [36], that is, they sequentially maximize the objective over subsets of elements of $\mathbf{x}$ and $\mathbf{s}$. The two other algorithms are clever heuristic procedures which do not possess any desirable "ascent" property but rather lead to a good compromise between performance and complexity. The four procedures introduced hereafter are respectively named Bayesian Matching Pursuit (BMP), Bayesian Orthogonal Matching Pursuit (BOMP), Bayesian Stagewise Orthogonal Matching Pursuit (BStOMP) and Bayesian Subspace Pursuit (BSP) because of the clear connections existing between them and their well-known "standard" counterparts: MP, OMP, StOMP and BSP. These connections will be emphasized and discussed in section V.

1) Bayesian Matching Pursuit (BMP): We define BMP as a block-ascent algorithm in which one single component j of $\hat{\mathbf{x}}^{(n)}$ and $\hat{\mathbf{s}}^{(n)}$ is modified at each iteration, that is

$$(\hat{\mathbf{x}}^{(n+1)},\hat{\mathbf{s}}^{(n+1)}) = \underset{(\mathbf{x},\mathbf{s})}{\arg\max} \log p(\mathbf{y},\mathbf{x},\mathbf{s}), \tag{19}$$

subject to $\forall i \neq j$:

$$\begin{aligned} x_i &= \hat{x}_i^{(n)}, \\ s_i &= \hat{s}_i^{(n)}. \end{aligned} \tag{20}$$

Since only the jth component varies between $(\hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)})$ and $(\hat{\mathbf{x}}^{(n+1)}, \hat{\mathbf{s}}^{(n+1)})$, the update (19)-(20) is completely characterized by the value of $(\hat{x}_j^{(n+1)}, \hat{s}_j^{(n+1)})$. We show in Appendix B that $\hat{x}_j^{(n+1)}$ and $\hat{s}_j^{(n+1)}$ can be expressed as

$$\hat{s}_j^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_j^{(n)}\mathbf{d}_j, \mathbf{d}_j \rangle^2 > T_j, \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

$$\hat{x}_j^{(n+1)} = \hat{s}_j^{(n+1)} \frac{\sigma_x^2}{\sigma_x^2 + \sigma_w^2} \left(\hat{x}_j^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle \right), \tag{22}$$

where

$$\mathbf{r}^{(n)} = \mathbf{y} - \mathbf{D}\hat{\mathbf{x}}^{(n)}, \tag{23}$$

$$T_j = 2\sigma_w^2 \frac{\sigma_x^2 + \sigma_w^2}{\sigma_x^2} \log\left(\frac{1-p_j}{p_j} \right). \tag{24}$$

A crucial question in the implementation of (19)-(20) is the choice of the index j to update at each iteration. For BMP, we choose to update the couple (x_j, s_j) leading to the maximum increase of $\log p(\mathbf{y},\mathbf{x},\mathbf{s})$, that is

$$j = \underset{k}{\arg\max} \{ \underset{(\mathbf{x},\mathbf{s})}{\max} \log p(\mathbf{y},\mathbf{x},\mathbf{s}) \}, \tag{25}$$

subject to (20) $\forall i \neq k$.

Equations (19)-(25) form the basis of the BMP algorithm. In order to particularize these recursions to the probabilistic model defined in section III, we define the following function:

$$\begin{aligned} \rho^{(n)}(x_i, s_i) = &- \| \mathbf{r}^{(n)} + (\hat{s}_i^{(n)}\hat{x}_i^{(n)} - s_i x_i)\mathbf{d}_i \|^2 \\ &- \epsilon\, x_i^2 - \lambda_i s_i, \end{aligned} \tag{26}$$

Initialization : $\hat{\mathbf{x}}^{(0)} = 0$, $\hat{\mathbf{s}}^{(0)} = 0$, $n = 0$.
Repeat :

1. Update the residual:

$$\mathbf{r}^{(n)} = \mathbf{y} - \mathbf{D}\hat{\mathbf{x}}^{(n)}.$$

2. Evaluate $\tilde{s}_i^{(n+1)}$ and $\tilde{x}_i^{(n+1)}$ as in (12)-(13).

3. Choose j as in (15)-(16).

4. Update the support and the coefficients:

$$\hat{s}_i^{(n+1)} = \begin{cases} \tilde{s}_i^{(n+1)} & \text{if } i = j, \\ \hat{s}_i^{(n)} & \text{otherwise,} \end{cases} \tag{30}$$

$$\begin{aligned} \hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n+1)}}^{(n+1)} &= \left(\mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}}^T \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}} + \tfrac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_{\|\hat{\mathbf{s}}^{(n+1)}\|_0}\right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}}^T \mathbf{y}, \\ \hat{x}_i^{(n+1)} &= 0 \quad \text{if } \hat{s}_i^{(n+1)} = 0. \end{aligned} \tag{31}$$

TABLE II
BOMP ALGORITHM

where $\epsilon = \sigma_w^2/\sigma_x^2$ and $\lambda_i = \sigma_w^2 \log((1-p_i)/p_i)$. $\rho^{(n)}(x_i, s_i)$ can be understood as the value of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ (up to some additive and multiplicative constants independent of x_i and s_i) when $(x_k, s_k) = (\hat{x}_k^{(n)}, \hat{s}_k^{(n)})\ \forall\, k \neq i$ (see (92)-(93) in Appendix B). Keeping this interpretation in mind and defining

$$(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}) = \underset{(x_i, s_i)}{\arg\max}\, \rho^{(n)}(x_i, s_i), \tag{27}$$

it is easy to see that (25) and (21)-(22) can respectively be rewritten as

$$j = \underset{i}{\arg\max}\, \rho^{(n)}(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}), \tag{28}$$

$$\begin{aligned} \hat{s}_j^{(n+1)} &= \tilde{s}_j^{(n+1)}, \\ \hat{x}_j^{(n+1)} &= \tilde{x}_j^{(n+1)}. \end{aligned} \tag{29}$$

Table I provides the analytical expressions of $\tilde{x}_i^{(n+1)}$ and $\tilde{s}_i^{(n+1)}$. The detailed derivations leading to these expressions are provided in Appendix B. A careful analysis of the operations described in Table I reveals that BMP has a complexity per iteration scaling as $\mathcal{O}(MN)$.

2) Bayesian Orthogonal Matching Pursuit (BOMP) : as BMP, we define BOMP as a particular instance of a block-coordinate ascent algorithm applied to (6). The subsets of variables with respect to which the objective is sequentially optimized differ however from those considered by BMP. In particular, we define

the BOMP recursions (by means of half iterations) as follows:

$$(\hat{\mathbf{x}}^{(n+\frac{1}{2})}, \hat{\mathbf{s}}^{(n+\frac{1}{2})}) = \underset{(\mathbf{x},\mathbf{s})}{\arg\max} \log p(\mathbf{y}, \mathbf{x}, \mathbf{s}), \tag{32}$$

subject to (20) $\forall i \neq j$, with j defined in (25), then

$$\hat{\mathbf{s}}^{(n+1)} = \hat{\mathbf{s}}^{(n+\frac{1}{2})}, \tag{33}$$

$$\hat{\mathbf{x}}^{(n+1)} = \underset{\mathbf{x}}{\arg\max} \log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}}^{(n+1)}). \tag{34}$$

BOMP is therefore a two-step procedure: in a first step BOMP optimizes the goal function with respect to a particular couple (x_j, s_j); this operation is strictly equivalent to BMP's recursion (19)-(20). In a second step, BOMP looks for the maximum of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ over $\mathbf{x}$ while $\mathbf{s} = \hat{\mathbf{s}}^{(n+1)} = \hat{\mathbf{s}}^{(n+\frac{1}{2})}$. The solution of this problem can be expressed as

$$\begin{aligned} \hat{\mathbf{x}}^{(n+1)}_{\hat{\mathbf{s}}^{(n+1)}} &= \left(\mathbf{D}^T_{\hat{\mathbf{s}}^{(n+1)}} \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}} + \tfrac{\sigma_w^2}{\sigma_x^2} \mathbf{I}_{\|\hat{\mathbf{s}}^{(n+1)}\|_0}\right)^{-1} \mathbf{D}^T_{\hat{\mathbf{s}}^{(n+1)}} \mathbf{y}, \\ \hat{x}_i^{(n+1)} &= 0 \quad \text{if } \hat{s}_i^{(n+1)} = 0. \end{aligned}$$

We refer the reader to Appendix B for a detailed derivation of this expression.

Particularizing (32)-(34) to the probabilistic model presented in section III, we obtain the implementation described in Table II. In our description, we used the fact that step (32) is strictly equivalent to (19)-(20) and can therefore be efficiently implemented as described in the previous section. Moreover, the value of $\hat{\mathbf{x}}^{(n+\frac{1}{2})}$ does not need to be explicitly evaluated since it is never used for the evaluation of $\hat{\mathbf{x}}^{(n+1)}$ and $\hat{\mathbf{s}}^{(n+1)}$. The crucial difference between BMP and BOMP lies in the coefficient update: (18) in BMP is replaced by (31) in BOMP; this alternative update has a larger computational load ($\mathcal{O}(\|\hat{\mathbf{s}}^{(n+1)}\|_0^3)$ for (31) against $\mathcal{O}(1)$ for (18)). The complexity per iteration of BOMP is therefore $\mathcal{O}(\|\hat{\mathbf{s}}^{(n+1)}\|_0^3 + MN)$.

3) Bayesian Stagewise Orthogonal Matching Pursuit (BStOMP) : We define BStOMP as a modified version of BOMP where several entries of the support vector $\mathbf{s}$ can be changed at each iteration. In particular, BStOMP is characterized by the following recursion:

$$\hat{s}_i^{(n+1)} = \tilde{s}_i^{(n+1)} \quad \forall\ i, \tag{35}$$

$$\hat{\mathbf{x}}^{(n+1)} = \underset{\mathbf{x}}{\arg\max} \log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}}^{(n+1)}). \tag{36}$$

where $\tilde{s}_i^{(n+1)}$ has been defined in (27). We remind the reader that $\tilde{s}_i^{(n+1)}$ corresponds to the optimal decision on s_i when $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ is optimized over (x_i, s_i) while $(x_k, s_k) = (\hat{x}_k^{(n)}, \hat{s}_k^{(n)})\ \forall k \neq i$. $\tilde{s}_i^{(n+1)}$

Initialization : $\hat{\mathbf{x}}^{(0)} = 0$, $\hat{\mathbf{s}}^{(0)} = 0$, $n = 0$.
Repeat :

1. Update the residual:

$$\mathbf{r}^{(n)} = \mathbf{y} - \mathbf{D}\hat{\mathbf{x}}^{(n)}.$$

2. Evaluate $\tilde{s}_i^{(n+1)}$ as in (12)

3. Update the support and the coefficients:

$$\hat{s}_i^{(n+1)} = \tilde{s}_i^{(n+1)} \quad \forall\, i, \tag{37}$$

$$\begin{aligned} \hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n+1)}}^{(n+1)} &= \left(\mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}}^T \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}} + \tfrac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_{\|\hat{\mathbf{s}}^{(n+1)}\|_0}\right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}}^T \mathbf{y}, \\ \hat{x}_i^{(n+1)} &= 0 \quad \text{if } \hat{s}_i^{(n+1)} = 0. \end{aligned} \tag{38}$$

TABLE III
BSTOMP ALGORITHM

can therefore be understood as the locally-optimal decision on s_i given the current estimate $(\hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)})$. Hence, in a nutshell, the philosophy behind BStOMP consists in setting each element of $\hat{\mathbf{s}}^{(n+1)}$ to its locally-optimal value given the current estimate $(\hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)})$. The update of $\hat{\mathbf{x}}^{(n+1)}$ is the same as for BOMP.

The operations performed by BStOMP are summarized in Table III. The complexity per iteration of BStOMP is similar to BOMP, that is $\mathcal{O}(\|\hat{\mathbf{s}}^{(n+1)}\|_0^3 + MN)$. In fact, BOMP and BStOMP only differ in the support update step: whereas BOMP only sets one element of $\hat{\mathbf{s}}^{(n)}$ to its locally-optimal value (see (30)), BStOMP does so for all components of the new support estimate (see (37)). A consequence of update (37) is that BStOMP is no longer an ascent algorithm. Nevertheless, we will see in the empirical results presented in section VI that the support update implemented by BStOMP allows for a reduction of the number of iterations (and hence of the overall running time) required to find the sought sparse vector.

4) Bayesian Subspace Pursuit (BSP) : We define BSP as another heuristic procedure in which a limited number of atoms can be selected or deselected at each iteration. As previously, the choice of the atoms selected/deselected is made from a "local" perspective. More formally, let us define

$$\rho_i^{(n)}(s_i) \triangleq \max_{x_i} \rho^{(n)}(x_i, s_i). \tag{39}$$

Hence, $\rho_i^{(n)}(s_i)$ corresponds to the maximum of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ when optimized over x_i for a given value of s_i and for $(x_k, s_k) = (\hat{x}_k, \hat{s}_k)$ $\forall k \neq i$. Using this definition, we define BSP as the following two-step

procedure. First, the support estimate is updated as

$$\hat{\mathbf{s}}^{(n+\frac{1}{2})} = \underset{\mathbf{s}\in\mathcal{S}_P}{\arg\max} \sum_i \rho_i^{(n)}(s_i), \tag{40}$$

where $\mathcal{S}_P = \{\mathbf{s} \,|\, \|\mathbf{s} - \hat{\mathbf{s}}^{(n)}\|_0 \leq P\}$, and $\hat{\mathbf{x}}^{(n+\frac{1}{2})}$ is computed as in (31). Then, in a second step, the support estimate is modified according to

$$\hat{\mathbf{s}}^{(n+1)} = \underset{\mathbf{s}\in\mathcal{S}_K}{\arg\max} \sum_i \rho_i^{(n+\frac{1}{2})}(s_i), \tag{41}$$

where $\mathcal{S}_K = \{\mathbf{s} \,|\, \|\mathbf{s}\|_0 = K\}$ and the coefficient estimate $\hat{\mathbf{x}}^{(n+1)}$ is again computed from (31).

In a nutshell, update (40) consists in selecting/deselecting the (at most) P atoms leading to the best local increases of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ around $(\hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)})$. This operation can be interpreted as an intermediate between BOMP and BStOMP support updates. In particular, if $P = 1$ (resp. $P = M$) one recovers BMP/BOMP (resp. BStOMP) update (17) (resp. (35)). In a second step, BSP modifies the support on the basis of the local variations of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ around $(\hat{\mathbf{x}}^{(n+\frac{1}{2})}, \hat{\mathbf{s}}^{(n+\frac{1}{2})})$ with the constraint that the new support has exactly K non-zero elements.

We show in Appendix B that $\rho_i^{(n)}(s_i) = \rho^{(n)}(\tilde{x}_i(s_i), s_i)$ where

$$\tilde{x}_i(s_i) = s_i \frac{\sigma_x^2}{\sigma_x^2 + \sigma_w^2} \left(\hat{x}_i^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle \right). \tag{42}$$

Remember that $\rho^{(n)}(x_i, s_i)\ \forall\, i$ can be evaluated with a complexity $\mathcal{O}(MN)$. Moreover, we emphasize in Appendix B that solving (40)-(41) essentially requires the sorting of $L(\leq M)$ metrics depending on $\rho^{(n)}(s_i)$. The complexity associated to this operation scales as $\mathcal{O}(L \log L)$. Hence, the complexity per iteration of BSP is similar to BOMP and BStOMP.

5) Parameter estimation and adaptive threshold : we now discuss the implementation of the estimation of the noise variance σ_w^2 into the iterative process defined by the Bayesian pursuit algorithms. At each iteration, we can consider the following maximum-likelihood estimate

$$\left(\hat{\sigma_w^2}\right)^{(n)} = \underset{\sigma_w^2}{\arg\max} \log p(\mathbf{y}, \hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)}), \tag{43}$$

$$= N^{-1} \|\mathbf{r}^{(n-1)}\|^2. \tag{44}$$

This estimate can be included within the pursuit recursions defined in the previous subsections. From a practical point of view, the algorithms described in Table I to III then remain identical but, at each iteration, σ_w^2 is replaced by its current estimate $\left(\hat{\sigma_w^2}\right)^{(n)}$. In particular, BMP and BOMP remains ascent

algorithms since (43) defines an ascent operation.

It is illuminating to focus in more details on the impact of the estimation of the noise variance on the update of the support $\hat{\mathrm{s}}^{(n)}$. In particular, replacing σ_w^2 by its estimate (44) leads to the following expression for T_i:

$$T_i^{(n)} \triangleq 2\frac{\|\mathbf{r}^{(n)}\|^2}{N} \log\left(\frac{1-p_i}{p_i}\right) \frac{\sigma_x^2 + N^{-1}\|\mathbf{r}^{(n)}\|^2}{\sigma_x^2}. \tag{45}$$

The threshold therefore becomes a function of the number of iterations. Moreover, as $\sigma_x^2 \to \infty$, (45) tends to:

$$T_i^{(n)} \overset{\sigma_x^2\to\infty}{=} 2\frac{\|\mathbf{r}^{(n)}\|^2}{N} \log\left(\frac{1-p_i}{p_i}\right). \tag{46}$$

The threshold is then proportional to the residual energy; the proportionality factor depends on the occurrence probability of each atom. In practice, $T_i^{(n)}$ has therefore the following operational meaning: during the first iterations, the residual is large (and so is $T_i^{(n)}$), and only the atoms having a large correlation with $\mathbf{y}$ are likely to be included in the support; after a few iterations, the norm of the residual error decreases and atoms weighted by smaller coefficients can enter the support.

V. Connections with previous works

The derivation of practical and effective algorithms searching for a solution of the sparse problem has been an active field of research for several decades. In order to properly replace our work in the ever-growing literature pertaining to this topic, we provide hereafter a short survey of some significant works in the domain. Note that, although our survey will necessarily be incomplete, we attempted to present the works the most connected with the proposed methodologies. In the first two subsections, we review the procedures belonging to the family of pursuit and Bayesian algorithms, respectively. In the last subsection, we emphasize some nice connections existing between the proposed procedures and some well-known pursuit algorithms of the literature, namely MP, OMP, StOMP and SP.

A. Pursuit algorithms

The designation "pursuit algorithms" generally refers to procedures looking for a sparse vector minimizing a goal function (most often the residual error $\mathbf{r}^{(n)}$) by making a succession of locally-optimal decisions on the support. The family of pursuit algorithms has a long history which traces back to 60's, for instance in the field of statistical regression [37].

Within this family, one can distinguish between *forward*, *backward* and *forward/backward* procedures. *Forward* algorithms gradually increase the support by sequentially *adding* new atoms. In this family, one can mention matching pursuit (MP) [12], orthogonal matching pursuit (OMP) [13], stagewise OMP (StOMP) [14], orthogonal least square (OLS) [15] or gradient pursuit (GP) [16]. These algorithms essentially differ in the way they select the atoms to be included in the support and/or the way they update the value of the non-zero coefficients.

Backward algorithms use the opposite strategy: they start from a support containing all the atoms of the dictionary and reduce it by sequentially removing "irrelevant" atoms. Backward algorithms have been extensively studied for undercomplete dictionaries in the statistical regression community [37]. They have been revisited more recently by Couvreur *et al.* in [38]. They are however of poor interest in overcomplete settings since most of them cannot make any relevant decision as soon as $N < M$.

Finally, *forward/backward* algorithms make iteratively a new decision on the support of the sparse vector by *either* adding and/or removing atoms from the current support. The first *forward/backward* algorithm we are aware of is due to Efroymson [39] and was placed in the context of statistical regression in undercomplete dictionaries. In his paper, the author suggested to add (resp. remove) *one* atom from the support if the decision leads to a residual error above (resp. below) a prespecified threshold. The choice of the threshold derives from considerations based on statistical hypothesis testing. Variations on this idea has been proposed in [40], [41] where the authors suggest different testing approaches.

Efroymson's procedure has later on been revisited in the context of sparse representations in overcomplete dictionary, see *e.g.*, [42], [43]. Other procedures, more flexible in the number of atoms added or removed from the support have been recently published. Let us mention the iterative hard thresholding (IHT) [17], hard thresholding pursuit (HTP) [18], compressive sampling matching pursuit (CoSaMP) [19] and subspace pursuit (SP) [20].

The procedures derived in section IV can be cast within the family of forward-backward pursuit algorithms since they build their estimate by a sequence of locally-optimal decisions and allow for both atom selection and deselection. However, unlike the proposed algorithms, most of the forward-backward procedures (*e.g.*, [17], [18], [20], [39], [40]) do not derive from an optimization problem but are rather clever heuristic methodologies. Moreover, the Bayesian framework in which the proposed methods arise can account for different prior information on the atom occurrence and encompass the estimation of some unknown model parameters. This is in contrast with the deterministic settings from which standard forward/backward algorithms derive.

B. Bayesian algorithms

Apart from some noticeable exceptions (*e.g.*, [6]), the development of sparse representation algorithms based on Bayesian methodologies seems to be more recent. The Bayesian algorithms available in the literature mainly differ in three respects: *i)* the probabilistic model they use to enforce sparsity; *ii)* the Bayesian criterion they intend to optimize (*e.g.*, minimum mean square error (MMSE), maximum a posteriori (MAP), etc.); *iii)* the practical procedure they apply to compute or approximate the sought solution (*e.g.*, gradient algorithm, variational approximation, Markov-Chain Monte-Carlo methods, etc.). Regarding the choice of the prior, a popular approach consists in modelling $\mathbf{x}$ as a continuous random variable whose distribution has a sharp peak to zero and heavy tails (*e.g.*, Laplace, t-Student or Jeyffrey's distributions). Such a strategy has been exploited, considering different Bayesian criteria and optimization strategies, in the following contributions [21]–[26]. Another approach, recently gaining in popularity, is based on a prior made up of the combination of Bernoulli and Gaussian distributions, see *e.g.*, [6], [7], [27]–[34]. Different variants of Bernoulli-Gaussian (BG) models exist. A first approach consists in modelling the elements of $\mathbf{x}$ as Gaussian variables whose variance is controlled by a Bernoulli variable:

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{D}\mathbf{x}, \sigma_w^2 \mathbf{I}_N), \tag{47}$$

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{M} p(x_i|s_i), \qquad p(\mathbf{s}) = \prod_{i=1}^{M} p(s_i), \tag{48}$$

where

$$p(x_i|s_i) = \mathcal{N}(0, \sigma_x^2(s_i)), \qquad p(s_i) = \mathrm{Ber}(p_i). \tag{49}$$

Hence a small variance $\sigma_x^2(0)$ enforces x_i to be close to zero if $s_i = 0$. Another model based on BG variables is (3)-(5), as considered in the present paper. Note that although models (3)-(5) and (47)-(49) are usually both referred to as "Bernoulli-Gaussian", they lead to different joint probabilistic models $p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ and therefore to different methodologies.

For a given BG model, the algorithms of the literature differ in the choice of the optimization criterion and the practical implementation they consider. Contributions [27]–[31] are based on model (47)-(49). In [27], the authors attempt to compute an (approximate) MMSE estimate of $\mathbf{x}$. To do so, they propose a heuristic procedure to identify the set of supports having the largest posterior probabilities $p(\mathbf{s}|\mathbf{y})$. In [28], the authors derived the so-called "Fast Bayesian Matching Pursuit" (FBMP) following a very similar approach. This contribution essentially differs from [27] in the way the set of supports with the

largest posterior probabilities $p(\mathbf{s}|\mathbf{y})$ is selected. The approach considered in [29] is based on the joint maximization of $p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ for the decoding of real-field codes. More particularly, the authors consider a relaxation of the Bernoulli distribution $p(\mathbf{s})$ and apply optimization techniques for smooth functions. The use of the sum-product algorithm [44] was investigated in [30] to compute approximation of the marginals $p(x_i|\mathbf{y})$. In the same spirit, another approach based on a mean-field approximation and the VB-EM algorithm [45], has been considered in [31] to derive approximate values of $p(\mathbf{x}|\mathbf{y})$, $p(\mathbf{s}|\mathbf{y})$ and $p(x_i, x_i|\mathbf{y})$. These approximate marginals are then used to make approximate MMSE or MAP decisions on $\mathbf{x}$ and $\mathbf{s}$. Finally, Ge *et al.* suggest in [46] another approximation of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$ based on a MCMC inference scheme.

On the other hand, model (3)-(5) has been considered in [6], [7], [33], [47]. Contribution [6] is the most related to the present work (in particular to BMP and BOMP): the authors proposed an ascent implementation of two MAP problems, involving either $p(\mathbf{y}, \mathbf{s})$ or $p(\mathbf{y}, \mathbf{x}, \mathbf{s})$. However, the subsets of variables over which the objective is maximized at each iteration differ from those considered in the implementation of the proposed BMP and BOMP. In [6], the authors focus on a particular application, namely the denoising of "geophysical signal" expressed in a wavelet basis, leading to a non-overcomplete setting. An extension to overcomplete settings has been considered by Soussen *et al.* in [7]. In [33], the authors focus on a MAP problem involving $p(s_i|\mathbf{y})$ to make a decision on the support of the sparse vector. The intractability of the MAP problem is addressed by means of a mean-field variational approximation. Finally, a different approach is considered in [47]: the authors make a decision on $\mathbf{s}$ by building a sequence of test of hypotheses; no particular Bayesian objective function is considered and the construction of the tests are, to some extent, clever but heuristic.

C. Connections with some well-known pursuit algorithms

In this section, we emphasize the connections existing between the procedures derived in section IV and some well-known standard pursuit procedures. In particular, we show that BMP, BOMP, BStOMP and BSP can be seen as forward/backward extension of MP, OMP, StOMP and SP for some particular values of the parameters of model (3)-(5).

We first show the equivalence of MP and BMP under the following setting: $\sigma_w^2 \to 0$, $\sigma_x^2 \to \infty$. Under

these conditions, BMP updates (12)-(13) can be rewritten as:

$$\tilde{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_i^{(n)} \mathbf{d}_i, \mathbf{d}_i \rangle^2 > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{50}$$

$$\tilde{x}_i^{(n+1)} = \tilde{s}_j^{(n+1)} \left(\hat{x}_i^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle \right). \tag{51}$$

First, note that any trace of σ_x^2 has disappeared in the resulting expressions; this intuitively makes sense since $\sigma_x^2 \to \infty$ corresponds to a non-informative prior on $\mathbf{x}$. More importantly, the inequality in the right-hand side of (50) is then "almost always" satisfied and therefore $\tilde{s}_i^{(n+1)} = 1 \ \forall i$. Indeed, $\tilde{s}_i^{(n+1)} = 0$ occurs if and only if

$$\hat{x}_i^{(n)} = -\langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle. \tag{52}$$

Now, assuming that $\langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle$ is a continuously-valued random variable (which usually makes sense in practice, especially in noisy scenarios), we have that (52) is satisfied with probability zero. Assuming then that $\tilde{s}_i^{(n+1)} = 1 \ \forall i$ and plugging (51) into (16), we obtain

$$\rho^{(n)}(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}) = \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2 - \|\mathbf{r}^{(n)}\|^2. \tag{53}$$

Finally, considering (51) and (53), BMP recursions can be summarized as

$$j = \arg\max_i \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2, \tag{54}$$

$$\hat{x}_i^{(n+1)} = \begin{cases} \hat{x}_i^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle & \text{if } i = j, \\ \hat{x}_i^{(n)} & \text{otherwise.} \end{cases} \tag{55}$$

Now, recursions (54)-(55) exactly correspond to the definition of MP [12]. Hence, in the particular case where $\sigma_w^2 \to 0$ and $\sigma_x^2 \to \infty$, BMP turns out to be equivalent to MP. In the general case, however, the two algorithms significantly differ since BMP implements features that are not available in MP. In particular, BMP implements atom deselection as soon as $\sigma_w^2 \neq 0$ whereas MP only allows for atom selection. Moreover, unlike MP, BMP can take into account some information about the atom occurrence (p_i's) or the variance of the non-zero coefficients (σ_x^2).

Similarly, OMP turns out to be a particular case of BOMP under the conditions $\sigma_w^2 \to 0$, $\sigma_x^2 \to \infty$. Indeed, first remind that the first step of BOMP (32) is strictly equivalent to BMP update (19)-(20).

Hence, from the discussion above we have that BOMP support update can be rewritten as

$$\hat{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } i = j, \\ \hat{s}_i^{(n)} & \text{otherwise,} \end{cases} \tag{56}$$

where j is defined in (54). Moreover, we have that

$$\lim_{\sigma_w^2 \to 0} \left(\mathbf{D}_{\mathbf{s}}^T \mathbf{D}_{\mathbf{s}} + \frac{\sigma_w^2}{\sigma_x^2} \mathbf{I}_{\|\mathbf{s}\|_0} \right)^{-1} \mathbf{D}_{\mathbf{s}}^T \mathbf{y} = \mathbf{D}_{\mathbf{s}}^{\dagger} \mathbf{y}. \tag{57}$$

Hence, BOMP update (31) becomes

$$\hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n+1)}}^{(n+1)} = \mathbf{D}_{\hat{\mathbf{s}}^{(n+1)}}^{\dagger} \mathbf{y}. \tag{58}$$

Now, (54), (56) and (58) correspond to the standard implementation of OMP [13].

Let us finally compare the recursions implemented by StOMP and BStOMP in the particular case where $\sigma_x^2 \to \infty$. Since BOMP and BStOMP (resp. OMP and StOMP) implement the same coefficient update, we only focus on the update of the support vector $\mathbf{s}$. First let us remind that StOMP support update is expressed as

$$\hat{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2 > \tilde{T}^{(n)}, \\ \hat{s}_i^{(n)} & \text{otherwise,} \end{cases} \tag{59}$$

where $\tilde{T}^{(n)}$ is a threshold which derives from hypothesis-testing considerations [14]. It is clear from (59) that StOMP is a forward algorithm; in particular, it selects at each iteration all atoms whose correlation with current residual exceeds a certain threshold.

On the other hand, if $\sigma_x^2 \to \infty$, BStOMP support update (37) can be rewritten as

$$\hat{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_i^{(n)} \mathbf{d}_i, \mathbf{d}_i \rangle^2 > T_i, \\ 0 & \text{otherwise.} \end{cases} \tag{60}$$

In particular, if the ith atom was not selected at iteration $n-1$, *i.e.*, $(\hat{x}_j^{(n)}, \hat{s}_j^{(n)}) = (0, 0)$, this expression becomes

$$\hat{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2 > T_i, \\ \hat{s}_i^{(n)} & \text{otherwise.} \end{cases} \tag{61}$$

Comparing (61) to (59), we see that the support update of StOMP and BStOMP are similar in such a case. However, in the general case (60), BStOMP allows for the deselection of atoms. Moreover, another crucial difference between StOMP and BStOMP lies in the definition of the threshold T_i and $\tilde{T}^{(n)}$.

Indeed, the Bayesian framework considered in this paper naturally leads to a definition of the threshold as a function of the model parameters. Unlike the approach followed in [14], it requires therefore no additional hypothesis and/or design criterion.

Finally, BSP reduces to SP as $\sigma_w^2 \to 0$, $\sigma_x^2 \to \infty$ and for certain modifications of the sets $\mathcal{S}_P$ and $\mathcal{S}_K$ appearing in (40) and (41). The proof of this connection is slightly more involved to establish and is reported to Appendix C.

VI. Simulation Results

In this section we illustrate the performance of the proposed algorithms. We evaluate the following metrics by Monte-Carlo simulation: *i)* the mean square error (MSE) on the non-zero coefficients of the sparse vector; *ii)* the probability of wrong decision on the support, *i.e.*, $P_e \triangleq p(\hat{s}_i \neq s_i)$. For each point of simulation, we averaged the results over 3000 trials.

A. The Uniform Case

We first consider the case where all the atoms have the same probability to be active, *i.e.*, $p_i = p \ \forall i$. For each experiment, the data vector $\mathbf{y}$ is generated according to model (3) with $\sigma_n^2 = 10^{-4}$ and $\sigma_x^2 = 1$. In Fig. 1 we represent the MSE, the probability of wrong decision on the elements of the support and the average running time achieved by different sparse-representation algorithms. Each point of simulation corresponds to a fixed number of non-zero coefficients, say K, and, given this number, the positions of the non-zero coefficients are drawn uniformly at random for each observation. We set $N = 154$, $M = 256$.

In addition to the proposed procedures, we consider several algorithms of the state of the art: MP [12], OMP [13], StOMP [14], SP [20], IHT [17], HTP [18], Basis Pursuit Denoising (BPD) [10], SBR [7], SOBAP [33] and FBMP [28]. The stopping criterion used for MP and OMP is based on the norm of the residual: the recursions are stopped as soon as the norm of the residual drops below $\sqrt{N\sigma_n^2}$. StOMP is run with the "CFAR" thresholding criterion [14]. BStOMP and BSP implement the noise variance estimation described in section IV-5. We set $p_i = \frac{K}{M}, \ \forall i$ in all the Bayesian pursuit algorithms.

We see in Fig. 1 that the Bayesian pursuit algorithms outperform their standard counterparts (MP, OMP, StOMP and SP) both in terms of MSE and probability of wrong detection of the support. The performance improvement depends upon the considered algorithms: whereas the gain brought by BMP, BStOMP and BSP is significant, BOMP only leads to a marginal improvement in terms of MSE. We also notice that the running times of the Bayesian and standard procedures are roughly similar.

The proposed Bayesian pursuit algorithms also perform well with respect to other algorithms of the literature. In particular, BSP and BStOMP are only outperformed by FBMP and SOBAP (resp. SOBAP) in terms of MSE (resp. probability of error on the support). The gain in performance allowed by FBMP and SOBAP has however to be contrasted with their complexity since they involve a much larger computational time.

We repeated similar experiments for many different values of N and K with $M = 256$, and we came to similar conlusions. Fig. 2 illustrates the results of these experiments in a "phase diagram": each curve represents the couples $(K/N, N/M)$ for which a particular algorithm achieves MSE=10^{-2} (top) or $P_e = 10^{-2}$ (bottom). Above (resp. below) these curves, the corresponding algorithms perform worse (resp. better). For the sake of readibility, we only reproduce the results for the proposed procedures and some of the algorithms considered in Fig. 1. As previously, we see that the Bayesian pursuit algorithms improve the performance with respect to their standard counterparts. The gain is the most significant for moderate-to-high value of N/M. We note the bad behavior of StOMP in terms of probability of error on the support: the algorithm achieved $P_e = 10^{-2}$ for no value of the parameters $(K/N, N/M)$. This is due to the fact that StOMP always selects too many atoms and can never removed them at subsequent iterations. In contrast, we see that the atom deselection process implemented by BStOMP solves this problem and leads to very good results in terms of support recovery.

Finally, we also assess the robustness of the proposed approaches to different levels of noise. Once again, we have observed that the conclusions drawn previously remain valid at lower signal-to-noise ratios. Fig. 3, which represents the MSE versus the noise variance, illustrates our purpose. We see that the Bayesian pursuit algorithms allow for an improvement of the performance irrespective of the noise level. The most effective procedures remain the more complex SoBaP and FBMP.

B. The Non-uniform Case

We now consider the case where the atoms of the dictionary have different probabilities to be active. We assume that the p_i's are independent realizations of a beta distribution $Beta(\alpha, \beta)$ with $\alpha = 0.4$, $\beta = 0.4$. The data are generated as previously. In particular, for each trial, the positions of the non-zero coefficients are drawn uniformly at random. This leads to a realization of $\mathbf{s}$. Knowing $\mathbf{s}$, we draw the value of p_i from its posterior distribution[2] $p(p_i|s_i)\ \forall\, i$.

In Fig. 4, we represent the performance of the Bayesian pursuit algorithms when they have access to

[2]It is easy to show that $p(p_i|s_i)$ is a beta distribution $Beta(\alpha', \beta')$ with $\alpha' = \alpha + s_i$ and $\beta' = \beta + 1 - s_i$.

the values of the occurrence probabilities p_i's. For the sake of comparison, we use the same simulation parameters as those used to generate Fig. 1. The red curves reproduce the performance of the Bayesian pursuit algorithms observed in Fig. 1. The blue curves illustrate the performance of the same algorithms when they are fed with the values of p_i's. We also show the algorithm performance when they are fed with noisy estimates, say $\hat{p}_i$'s, of the actual prior p_i (green curves) :

$$\hat{p}_i = p_i + \mathcal{U}(\max(0, p_i - \Delta_p), \min(1, p_i + \Delta_p)), \tag{62}$$

where $\mathcal{U}(a, b)$ denotes a uniform on $[a, b]$ and Δ_p is a parameter determining the level of the perturbation. We set $\Delta_p = 0.3$ in Fig. 4.

Since the p_i's constitute an additional source of information about the position of the non-zero coefficients in the sparse vector, the Bayesian pursuit algorithms are expected to improve the recovery performance. This is observed in Fig. 4: all the Bayesian algorithms achieve better performance when informed of the values of p_i's. The improvement of the performance of BMP and BOMP is only marginal. On the contrary, a significant gain of performance is observed for BStOMP and BSP, both in terms of MSE and probability of error on the support. We can also see that the algorithms are quite robust to an error on the a priori probabilities: the setup $\Delta_p = 0.3$ degrades but still allows improvements with respect to the non-informed case.

As in the uniform case, we repeated these experiments for many different values of K and N, with $M = 256$. This leads to the phase diagrams represented in Fig. 5. Each curve represents the set of couples $(K/N, N/M)$ for which a particular algorithm achieves MSE=10^{-2} (top) or $P_e = 10^{-2}$ (bottom). The curves obtained in Fig. 2 for the uniform case (that is using $p_i = \frac{K}{M},\ \forall i$ in the algorithms) are reproduced in red. The blue curves illustrate the performance of the algorithms informed with the values of p_i's. We see that the conclusions drawn from Fig. 4 are confirmed by the phase diagram: the exploitation of the prior information enables to enhance the algorithm performance. The most impressive improvements are obtained by BSP and BStOMP. We also note the good behavior of BOMP for large values of N/M.

VII. Conclusions

The contribution of this paper is three-fold. First, we emphasized a connection between the standard penalized sparse problem and a MAP problem involving a BG model. Although this model has been previously involved in several contributions, its link with the standard sparse representation problem had been poorly addressed so far. Only recently, some contributions proposed results in that vein. The result derived in this paper is however more general since it requires weaker conditions on the observed vector

$\mathbf{y}$.

Secondly, we proposed several methodologies to search for the solution of the considered MAP problem. The proposed algorithms turn out to be forward/backward pursuit algorithms, which account for the a priori information available on the atoms occurrences. We showed on different experimental setups the good behavior of the proposed procedures with respect to several algorithms of the literature.

Finally, we established interesting connections between the proposed procedures and well-known algorithms of the literature: MP, OMP, StOMP and SP. We believe that these connections build enlightening bridges between these standard procedures (which have been known, for some of them, for more than two decades now) and Bayesian procedures which have attracted the attention of the sparse-representation community more recently.

In our future work, we will focus on the derivation of guarantees for the recovery of the "true" support of the sparse decomposition. To the best of our knowledge, the derivation of such guarantees for forward/backward algorithms mostly remain an open (and difficult) problem in the literature. A few results have been proposed in particular settings, see *e.g.*, [19], [20]. However, no general guarantees exists for many (even very simple) forward/backward algorithms as [39]–[41].

APPENDIX A
PROOF OF THEOREM 1

In this appendix, we prove the equivalence between (1) and (6) under the hypotheses of Theorem 1, *i.e.*, *i)* $\|\mathbf{y}\|_2 < \infty$, *ii)* $\sigma_x^2 \rightarrow \infty$, *iii)* $p_i = p \ \forall i$, *iv)* $\lambda \triangleq 2\sigma_w^2 \log(\frac{1-p}{p})$. For the sake of clarity and conciseness, we restrict the demonstration to the case where any subset of $L \leq N$ columns of $\mathbf{D}$ is linearly independent. The general case can however be derived in a similar way.

We first pose some notations and definitions. Let

$$\mathcal{X}^\star \triangleq \arg\min_{\mathbf{x}} \{\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0\},$$

be the set of solutions of the standard sparse representation problem. We define $f(\mathbf{s})$ as

$$f(\mathbf{s}) \triangleq \min_{\mathbf{x}} \{\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0\} \quad \text{s.t. } x_i = 0 \ \text{ if } s_i = 0. \tag{63}$$

$f(\mathbf{s})$ is therefore the minimum of the standard SR problem (1) if the support of the sparse vector is fixed.

Moreover we define $\mathbf{x}^\star(\mathbf{s})$ as

$$\mathbf{x}^\star_{\mathbf{s}}(\mathbf{s}) \triangleq \mathbf{D}^\dagger_{\mathbf{s}}\mathbf{y}, \text{ and } x^\star_i(\mathbf{s}) \triangleq 0 \text{ if } s_i = 0. \tag{64}$$

Clearly, $\mathbf{x}^\star(\mathbf{s})$ is a solution of (63). It is the unique (resp. the minimum ℓ_2-norm) solution if $\|\mathbf{s}\|_0 \leq N$ (resp. $\|\mathbf{s}\|_0 > N$). Using these notations, we can redefine $\mathcal{X}^\star$ as follows

$$\mathcal{X}^\star = \{\mathbf{x} \in \mathbb{R}^M | \mathbf{x} = \mathbf{x}^\star(\mathbf{s}) \text{ with } \mathbf{s} \in \arg\min_{\mathbf{s}} f(\mathbf{s})\}. \tag{65}$$

This definition is valid because the minimum of $f(\mathbf{s})$ is necessarily achieved for $\mathbf{s}$ such that $\|\mathbf{s}\|_0 \leq N$, in which case (64) is the unique solution of (63).

Let moreover

$$g(\mathbf{s}, \sigma_x^2) \triangleq 2\sigma_w^2 \min_{\mathbf{x}}\{-\log p(\mathbf{y}, \mathbf{x}, \mathbf{s}) - \log(1-p)\}, \tag{66}$$

$$g(\mathbf{s}) \triangleq \lim_{\sigma_x^2 \to \infty} g(\mathbf{s}, \sigma_x^2), \tag{67}$$

$$\hat{\mathbf{x}}(\mathbf{s}) \triangleq \arg\min_{\mathbf{x}}\{-\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})\}. \tag{68}$$

The goal function in (66) is equal, up to an additive constant, to $-\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ and $g(\mathbf{s})$ corresponds to its minimum (over $\mathbf{x}$) when $\sigma_x^2 \to \infty$. The equality in (68) is well-defined since the minimum exists and is unique as shown below. With these notations, we can define the set of solutions of the BG problem (6) when $\sigma_x^2 \to \infty$ as

$$\hat{\mathcal{X}} \triangleq \{\mathbf{x} \in \mathbb{R}^M | \mathbf{x} = \hat{\mathbf{x}}(\mathbf{s}) \text{ with } \mathbf{s} \in \arg\min_{\mathbf{s}} g(\mathbf{s})\}. \tag{69}$$

We want therefore to show that $\mathcal{X}^\star = \hat{\mathcal{X}}$ under the hypotheses of Theorem 1.

The proof of Theorem 1 is based on the following intermediate results:

$$\lim_{\sigma_x^2 \to \infty} \hat{\mathbf{x}}(\mathbf{s}) = \mathbf{x}^\star(\mathbf{s}) \quad \forall \mathbf{s}, \tag{70}$$

$$g(\mathbf{s}) \geq f(\mathbf{s}) \quad \forall \mathbf{s}, \tag{71}$$

$$\{f(\mathbf{s}) | \, \mathbf{s} \in \{0,1\}^M\} \subseteq \{g(\mathbf{s}) | \, \mathbf{s} \in \{0,1\}^M\}, \tag{72}$$

$$\min_{\mathbf{s}} g(\mathbf{s}) = \min_{\mathbf{s}} f(\mathbf{s}), \tag{73}$$

$$\arg\min_{\mathbf{s}} g(\mathbf{s}) \subseteq \arg\min_{\mathbf{s}} f(\mathbf{s}). \tag{74}$$

1) Proof of (70)*:* From standard Bayesian theory (see for example [48, Chap. 14]), it can be seen that the unique minimum of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ is given by

$$\begin{aligned} &\hat{\mathbf{x}}_{\mathbf{s}}(\mathbf{s}) = \left(\mathbf{D}_{\mathbf{s}}^T\mathbf{D}_{\mathbf{s}} + \tfrac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_k\right)^{-1}\mathbf{D}_{\mathbf{s}}^T\mathbf{y}, \\ &\hat{x}_i(\mathbf{s}) = 0 \qquad\qquad\qquad\qquad\qquad \text{if } s_i = 0. \end{aligned} \tag{75}$$

Taking the limit for $\sigma_x^2 \to \infty$, we obtain the equivalence between (64) and (75).

2) Proof of (71)*:* Using (70) and taking (3)-(5) into account, we have

$$\begin{aligned} g(\mathbf{s}) = \|\mathbf{y} - \mathbf{D}\mathbf{x}^\star(\mathbf{s})\|_2^2 - 2\sigma_w^2 \log p(\mathbf{s}) - 2\sigma_w^2 \log(1-p) \\ + \sigma_w^2 \lim_{\sigma_x^2 \to \infty} \frac{\|\hat{\mathbf{x}}(\mathbf{s})\|_2^2}{\sigma_x^2}. \end{aligned} \tag{76}$$

Since $\|\mathbf{y}\|_2 < \infty$ by hypothesis, it follows that

$$\lim_{\sigma_x^2 \to \infty} \|\hat{\mathbf{x}}(\mathbf{s})\|_2 = \|\mathbf{x}^\star(\mathbf{s})\|_2 = \|\mathbf{D}_{\mathbf{s}}^\dagger \mathbf{y}\|_2 < \infty, \tag{77}$$

since $\mathbf{D}_{\mathbf{s}}^\dagger$ is a bounded operator. Hence, the last term in (76) tends to zero. Moreover, since $p_i = p \ \forall i$ one can rewrite $p(\mathbf{s})$ as

$$\log p(\mathbf{s}) = -\|\mathbf{s}\|_0 \log\left(\frac{1-p}{p}\right) + \log(1-p). \tag{78}$$

Therefore, letting $\lambda \triangleq 2\sigma_w^2 \log(\frac{1-p}{p})$, we obtain

$$g(\mathbf{s}) = \|\mathbf{y} - \mathbf{D}\mathbf{x}^\star(\mathbf{s})\|_2^2 + \lambda\|\mathbf{s}\|_0. \tag{79}$$

Finally, realizing that

$$\|\mathbf{x}^\star(\mathbf{s})\|_0 \leq \|\mathbf{s}\|_0 \quad \forall \mathbf{s}, \tag{80}$$

we come up with (71). It is interesting to note that the equality holds in (71) if and only if $\|\mathbf{x}^\star(\mathbf{s})\|_0 = \|\mathbf{s}\|_0$, *i.e.*, when $x_i^\star(\mathbf{s}) \neq 0 \ \forall s_i = 1$.

3) Proof of (72)*:* To prove this result, we show that $\forall \mathbf{s} \in \{0,1\}^M$, $\exists\, \tilde{\mathbf{s}} \in \{0,1\}^M$ such that $f(\mathbf{s}) = g(\tilde{\mathbf{s}})$. First, notice that the value of $f(\mathbf{s})$ is only a function of $\mathbf{x}^\star(\mathbf{s})$. Now, $\forall \mathbf{s}$ one can find $\tilde{\mathbf{s}}$ such that

$$\mathbf{x}^\star(\mathbf{s}) = \mathbf{x}^\star(\tilde{\mathbf{s}}), \tag{81}$$

$$\|\mathbf{x}^\star(\tilde{\mathbf{s}})\|_0 = \|\tilde{\mathbf{s}}\|_0. \tag{82}$$

In order to see the last assertion, define $\tilde{\mathbf{s}}$ as

$$\tilde{s}_i = \begin{cases} 1 & \text{if } x_i^\star(\mathbf{s}) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{83}$$

Clearly, $\mathbf{s}$ and $\tilde{\mathbf{s}}$ differ as soon as $\|\mathbf{x}^\star(\mathbf{s})\|_0 \neq \|\mathbf{s}\|_0$. Considering problem (63), $\mathbf{s}$ and $\tilde{\mathbf{s}}$ introduce therefore different sets of constraints on the solution. By definition, the contraints defined by $\tilde{\mathbf{s}}$ include those defined by $\mathbf{s}$. However, the new constraints introduced by $\tilde{\mathbf{s}}$ has no effect on the solution since they correspond to $x_i^\star(\mathbf{s}) = 0$. Then clearly, (81) and (82) follow. Finally, (81) ensures that $f(\mathbf{s}) = f(\tilde{\mathbf{s}})$ and (82) implies $f(\tilde{\mathbf{s}}) = g(\tilde{\mathbf{s}})$ as emphasized in the remark below (80). This shows (72).

4) Proof of (73) *and* (74)*:* Let $\mathbf{s}^\star$ and $\hat{\mathbf{s}}$ be minimizers of $f(\mathbf{s})$ and $g(\mathbf{s})$ respectively. Then, we have

$$\begin{aligned} f(\mathbf{s}^\star) &\overset{(a)}{=} g(\tilde{\mathbf{s}}) \text{ for some } \tilde{\mathbf{s}}, \\ &\overset{(b)}{\geq} g(\hat{\mathbf{s}}) \overset{(c)}{\geq} f(\hat{\mathbf{s}}) \overset{(d)}{\geq} f(\mathbf{s}^\star), \end{aligned} \tag{84}$$

where *(a)* is a consequence of (72); *(c)* results from (71) and *(b)*, *(d)* follow from the definition of $\mathbf{s}^\star$ and $\hat{\mathbf{s}}$. Looking at the right and left-most terms, we see that the equality holds throughout (84). Hence, (73) results from $f(\mathbf{s}^\star) = g(\hat{\mathbf{s}})$ and (74) from $f(\mathbf{s}^\star) = f(\hat{\mathbf{s}})$.

5) Proof of $\hat{\mathcal{X}} = \mathcal{X}^\star$*:* First, it is easy to see that the inclusion $\hat{\mathcal{X}} \subseteq \mathcal{X}^\star$ holds by considering the definition of $\mathcal{X}^\star$ and $\hat{\mathcal{X}}$ (in (65) and (69)) and the technical results (70) and (74).

The reverse inclusion ($\mathcal{X}^\star \subseteq \hat{\mathcal{X}}$) can be proved by showing that $\forall\, \mathbf{s}^\star \in \arg\min_{\mathbf{s}} f(\mathbf{s}), \exists\, \hat{\mathbf{s}} \in \arg\min_{\mathbf{s}} g(\mathbf{s})$ such that

$$\mathbf{x}^\star(\mathbf{s}^\star) = \lim_{\sigma_x^2 \to \infty} \hat{\mathbf{x}}(\hat{\mathbf{s}}). \tag{85}$$

For any $\mathbf{s}^\star$, let us define $\hat{\mathbf{s}}$ as

$$\hat{s}_i = \begin{cases} 0 & \text{if } x_i^\star(\mathbf{s}^\star) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{86}$$

and let us show that such an $\hat{\mathbf{s}}$ satisfies the above conditions. From this definition, we have that

$$\|\mathbf{x}^\star(\hat{\mathbf{s}})\|_0 = \|\hat{\mathbf{s}}\|_0, \tag{87}$$

$$\mathbf{x}^\star(\mathbf{s}^\star) = \mathbf{x}^\star(\hat{\mathbf{s}}). \tag{88}$$

Hence, combining (70) with the last equality, we obtain (85). It thus remains to show that $\hat{\mathbf{s}}$ is a minimizer

of $g(\mathbf{s})$. This can be seen from the following sequence of equalities:

$$f(\mathbf{s}^\star) \overset{(a)}{=} f(\hat{\mathbf{s}}) \overset{(b)}{=} g(\hat{\mathbf{s}}) \overset{(c)}{=} \min_{\mathbf{s}} g(\mathbf{s}), \tag{89}$$

where *(a)* follows from (88); *(b)* is a consequence of (87); and *(c)* results from (73) and the definition of $\mathbf{s}^\star$.

APPENDIX B
DERIVATION OF BAYESIAN PURSUIT ALGORITHMS

A. BMP

Let us define

$$\hat{\mathbf{x}}_i^{(n)} \triangleq [\hat{x}_1^{(n)} \ \ldots \ \hat{x}_{i-1}^{(n)} \ x_i \ \hat{x}_{i+1}^{(n)} \ \ldots \ \hat{x}_M^{(n)}]^T, \tag{90}$$

$$\hat{\mathbf{s}}_i^{(n)} \triangleq [\hat{s}_1^{(n)} \ \ldots \ \hat{s}_{i-1}^{(n)} \ s_i \ \hat{s}_{i+1}^{(n)} \ \ldots \ \hat{s}_M^{(n)}]^T, \tag{91}$$

and

$$\rho^{(n)}(x_i, s_i) \triangleq \sigma_w^2 \log p(\mathbf{y}, \hat{\mathbf{x}}_i^{(n)}, \hat{\mathbf{s}}_i^{(n)}). \tag{92}$$

Using the BG model defined in III, we obtain

$$\begin{aligned} \rho^{(n)}(x_i, s_i) &= \sigma_w^2 \Big(\log p(\mathbf{y}|\hat{\mathbf{x}}_i^{(n)}, \hat{\mathbf{s}}_i^{(n)}) + \log p(\hat{\mathbf{x}}_i^{(n)}) + \log p(\hat{\mathbf{s}}_i^{(n)}) \Big), \\ &\propto - \|\mathbf{r}^{(n)} + (\hat{s}_i^{(n)} \hat{x}_i^{(n)} - s_i x_i)\mathbf{d}_i\|^2 - \epsilon \, x_i^2 - \lambda_i s_i, \end{aligned} \tag{93}$$

where $\epsilon = \sigma_w^2/\sigma_x^2$ and $\lambda_i = \sigma_w^2 \log((1-p_i)/p_i)$.

Then, clearly (19)-(20) and (25) are tantamount to solving

$$(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}) = \arg\max_{(x_i, s_i)} \rho^{(n)}(x_i, s_i) \quad \forall \, i, \tag{94}$$

$$j = \arg\max_i \rho^{(n)}(\tilde{x}_i^{(n+1)}, \tilde{s}_i^{(n+1)}), \tag{95}$$

and setting $\hat{s}_i^{(n+1)}$ and $\hat{x}_i^{(n+1)}$ as in (17) and (18).

Let us then derive the analytical expression of the solution of (94). First, we have

$$\begin{aligned}\tilde{x}_i(s_i) &\triangleq \arg\max_{x_i} \rho^{(n)}(x_i, s_i),\\ &= s_i \frac{\sigma_x^2}{\sigma_x^2+\sigma_w^2}\left(\hat{x}_i^{(n)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_i\rangle\right).\end{aligned} \tag{96}$$

Indeed, since $\rho^{(n)}(x_i, s_i)$ is a convex function of x_i, the last equality is simply obtained by solving $\frac{\partial \rho^{(n)}(x_i,s_i)}{\partial x_i} = 0$. Moreover, by definition, $\tilde{s}_i^{(n+1)}$ can be expressed as

$$\tilde{s}_i^{(n+1)} = \begin{cases} 1 & \text{if } \rho^{(n)}(\tilde{x}_i(1),1) > \rho^{(n)}(\tilde{x}_i(0),0),\\ 0 & \text{otherwise,}\end{cases} \tag{97}$$

Using the definitions of $\rho^{(n)}(x_i, s_i)$ and $\tilde{x}_i(s_i)$, the inequality in the right-hand side of (97) is equivalent to

$$\langle \mathbf{r}^{(n)} + \hat{x}_i^{(n)}\mathbf{d}_i, \mathbf{d}_i\rangle^2 > 2\sigma_w^2 \frac{\sigma_x^2+\sigma_w^2}{\sigma_x^2} \log\left(\frac{1-p_i}{p_i}\right). \tag{98}$$

Combining, (96), (97) and (98), we recover (11)-(14).

B. BOMP

The first step of BOMP (32) has been discussed in the previous subsection. We therefore focus on (34). We have that

$$\log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}}) \propto -\|\mathbf{y} - \mathbf{D}_{\hat{\mathbf{s}}}\mathbf{x}_{\hat{\mathbf{s}}}\|^2 - \frac{\sigma_w^2}{\sigma_x^2}\|\mathbf{x}\|^2. \tag{99}$$

First, if $\hat{s}_i = 0$, the corresponding element x_i only appears in the second term in the right-hand side of (99), and $x_i = 0$ clearly maximizes (99). It thus remains to find the value of $\mathbf{x}_{\hat{\mathbf{s}}}$. Since $\log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}})$ is a strictly concave function of $\mathbf{x_s}$, the optimal value of $\mathbf{x_s}$ can be found by solving $\nabla_{\mathbf{x_s}} \log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}}) = 0$ where

$$\nabla_{\mathbf{x_s}} \log p(\mathbf{y}, \mathbf{x}, \hat{\mathbf{s}}) = -2\mathbf{D}_{\hat{\mathbf{s}}}^T\mathbf{y} - 2(\mathbf{D}_{\hat{\mathbf{s}}}^T\mathbf{D}_{\hat{\mathbf{s}}} + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I})\,\mathbf{x}_{\hat{\mathbf{s}}}.$$

Finally, noticing that $\mathbf{D}_{\hat{\mathbf{s}}}^T\mathbf{D}_{\hat{\mathbf{s}}} + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I}$ is definite positive (and therefore invertible), we obtain (31).

C. BSP

The relation $\rho_i^{(n)}(s_i) = \rho^{(n)}(\tilde{x}_i(s_i), s_i)$ directly follows from (39) and (96). Let us then elaborate on maximizations (40)-(41). The goal function of (40)-(41) can be expressed as[3]

$$\sum_i \rho_i(s_i) \propto \sum_i \Delta(s_i) \triangleq h(\mathbf{s}),$$

where $\Delta(s_i) \triangleq \rho_i(s_i) - \rho_i(\hat{s}_i^{(n)})$. Let $\mathbf{s}$ be such that $s_i \neq \tilde{s}_i$ for $i \in \mathcal{I}$, where $\tilde{\mathbf{s}}$ is defined as in (12); $h(\mathbf{s})$ can be expressed as

$$h(\mathbf{s}) = \sum_i \Delta(\tilde{s}_i) - \left(\sum_{i \in \mathcal{I}} \Delta(\tilde{s}_i) - \Delta(1 - \tilde{s}_i) \right). \tag{100}$$

Now, by definition of $\tilde{s}_i$, we have $\Delta(\tilde{s}_i) \geq 0$, $\Delta(1 - \tilde{s}_i) \leq 0$. Hence, the generic solution of (40)-(41) can be written as

$$\hat{s}_i^{(n+1)} = \begin{cases} 1 - \tilde{s}_i & \text{if } i \in \mathcal{I}^\star, \\ \tilde{s}_i & \text{otherwise,} \end{cases} \tag{101}$$

where

$$\mathcal{I}^\star = \arg\min_{\mathcal{I}} \sum_{i \in \mathcal{I}} \Delta(\tilde{s}_i) - \Delta(1 - \tilde{s}_i), \tag{102}$$

subject to some constraints on $\mathcal{I}$ depending on the considered problem. For instance, let us particularize the constraints on $\mathcal{I}$ associated to (41): if $\|\tilde{\mathbf{s}}\|_0 \geq K$, then the definition of $\mathcal{S}_K$ implies that $\mathcal{I}$ must be such that

$$\begin{aligned} &\tilde{s}_i = 1 \qquad \forall\, i \in \mathcal{I}, \\ &\mathrm{Card}(\mathcal{I}) = \|\tilde{\mathbf{s}}\|_0 - K. \end{aligned} \tag{103}$$

Similarly, if $\|\tilde{\mathbf{s}}\|_0 < K$, the constraints become

$$\begin{aligned} &\tilde{s}_i = 0 \qquad \forall\, i \in \mathcal{I}, \\ &\mathrm{Card}(\mathcal{I}) = K - \|\tilde{\mathbf{s}}\|_0. \end{aligned} \tag{104}$$

Hence, solving (102) subject to (103) (resp. (104)) is equivalent to identifying the $\|\tilde{\mathbf{s}}\|_0 - K$ (resp. $K - \|\tilde{\mathbf{s}}\|_0$) indices with $\tilde{s}_i = 1$ (resp. $\tilde{s}_i = 0$) leading to the smallest metrics $\Delta(\tilde{s}_i) - \Delta(1 - \tilde{s}_i)$. Now, $\Delta(\tilde{s}_i) - \Delta(1 - \tilde{s}_i) = \rho_i(\tilde{s}_i) - \rho_i(1 - \tilde{s}_i)$ and $\rho_i(s_i)$ can be evaluated efficiently as described above.

[3]We drop the iteration index for conciseness.

Problem (40) can be solved in a similar way and is not further described herefater.

APPENDIX C
CONNECTION BETWEEN BSP AND SP

In this section, we emphasize the connection existing between BSP and SP when $\sigma_w^2 \to 0$, $\sigma_x^2 \to \infty$. For the sake of clarity, we first briefly remind SP implementation. SP operates in two steps. First, P atoms are added to the previous support as follows:

$$\hat{\mathbf{s}}^{(n+\frac{1}{2})} = \underset{\mathbf{s}\in\tilde{\mathcal{S}}_P}{\arg\max} \sum_i s_i \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2 \tag{105}$$

where

$$\tilde{\mathcal{S}}_P \triangleq \{\mathbf{s}|\, \|\mathbf{s} - \hat{\mathbf{s}}^{(n)}\|_0 = P, s_i = 1 \text{ if } s_i^{(n)} = 1 \;\forall\, i\}. \tag{106}$$

Then, $\hat{\mathbf{x}}^{(n+\frac{1}{2})}$ is evaluated according to (58). In a second step, SP removes some atoms from the support by applying the following rule:

$$\hat{\mathbf{s}}^{(n+1)} = \underset{\mathbf{s}\in\tilde{\mathcal{S}}_K}{\arg\max} \sum_i s_i \, (\hat{x}_i^{(n+\frac{1}{2})})^2 \tag{107}$$

where

$$\tilde{\mathcal{S}}_K \triangleq \{\mathbf{s}|\, \|\mathbf{s}\|_0 = K, s_i = 0 \text{ if } \hat{s}_i^{(n+\frac{1}{2})} = 0 \;\forall\, i\}. \tag{108}$$

$\hat{\mathbf{x}}^{(n+1)}$ is evaluated from (58).

We next show that BSP is equivalent to SP if we replace $\mathcal{S}_P$ by $\tilde{\mathcal{S}}_P$ in (40), $\mathcal{S}_K$ by $\tilde{\mathcal{S}}_K$ in (41), and let $\sigma_w^2 \to 0$, $\sigma_x^2 \to \infty$. We already showed the equivalence between the coefficient updates (31) and (58) in the previous section. It thus remains to prove the equivalence between (105) and (40) (resp. (107) and (41)). In order to do so, we use the following relation:

$$\rho_i^{(n)}(1) - \rho_i^{(n)}(0) = \begin{cases} \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2 & \text{if } \hat{s}_i^{(n)} = 0 \\ -(\hat{x}_i^{(n)})^2 & \text{if } \hat{s}_i^{(n)} = 1 \end{cases} \tag{109}$$

which can easily be proved by exploiting (13) and (16). Details are not reported here.

Setting $\mathcal{S}_P = \tilde{\mathcal{S}}_P$, problem (40) can also be equivalently rewritten as

$$\begin{aligned}\hat{\mathbf{s}}^{(n+\frac{1}{2})} &= \arg\max_{\mathbf{s}\in\tilde{\mathcal{S}}_P} \sum_i (\rho_i^{(n)}(s_i) - \rho_i^{(n)}(0)),\\ &= \arg\max_{\mathbf{s}\in\tilde{\mathcal{S}}_P} \sum_i s_i \langle \mathbf{r}^{(n)}, \mathbf{d}_i\rangle^2 \end{aligned} \tag{110}$$

where the last equality follows from (109). Similarly, setting $\mathcal{S}_K = \tilde{\mathcal{S}}_K$, problem (41) can be expressed as

$$\begin{aligned}\hat{\mathbf{s}}^{(n+1)} &= \arg\max_{\mathbf{s}\in\tilde{\mathcal{S}}_K} \sum_i (\rho_i^{(n)}(s_i) - \rho_i^{(n)}(1)),\\ &= \arg\max_{\mathbf{s}\in\tilde{\mathcal{S}}_K} -\sum_i (1-s_i)\,(\hat{x}_i^{(n+\frac{1}{2})})^2,\\ &= \arg\max_{\mathbf{s}\in\tilde{\mathcal{S}}_K} \sum_i s_i\,(\hat{x}_i^{(n+\frac{1}{2})})^2 \end{aligned} \tag{111}$$

where the last equalities are a consequence of (109). Comparing (110) with (105) (resp. (111) with (107)) we obtain the result.

REFERENCES

[1] Alan Miller, *Subset Selection in Regression, Second Editon*, Chapman and Hall/CRC, 2 edition, Apr. 2002.

[2] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[3] B. D. Jeffs and M. Gunsay, "Restoration of blurred star field images by maximally sparse optimization," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 202–211, Apr. 1993.

[4] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: an adaptative thresholding strategy," *IEEE Trans. on Image Process.*, vol. 16, no. 11, pp. 2675–2681, Nov. 2007.

[5] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, Apr. 1995.

[6] John J. Kormylo and Jerry M. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Transactions on Information Theory*, vol. 28, pp. 482–488, 1982.

[7] C. Soussen, J. Idier, D. Brie, and J. Duan, "From bernoulli-gaussian deconvolution to sparse signal restoration," *Signal Processing, IEEE Transactions on*, vol. 59, no. 10, pp. 4572–4584, Oct. 2011.

[8] I. Barbu, C. Herzet, and E. Mémin, "Sparse models and pursuit algorithms for piv tomography," in *Forum on recent developments in Volume Reconstruction techniques applied to 3D fluid and solid mechanics*, Nov. 2011.

[9] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[10] S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by Basis Pursuit," *SIAM J. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1999.

[11] I. Gorodnitsky and D. R. Bhaskar, "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.

[12] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 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 Ann. Asilomar Conf. Signals, Systems, and Computers*, 1993.

[14] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," 2006.

[15] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.

[16] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2370–2382, June 2008.

[17] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, Dec. 2008.

[18] S. Foucart, "Hard thresholding pursuit: an algorithm for compressive sensing," 2011.

[19] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 301–321, 2009.

[20] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," Jan. 2009.

[21] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[22] M. Girolami, "A variational method for learning sparse and overcomplete representations," *Neural Computation*, vol. 13, no. 11, pp. 2517–2532, 2001.

[23] C. Févotte and S. J. Godsill, "Blind separation of sparse sources using Jeffrey's inverse prior and the EM algorithm," in *Proc. 6th International Conference on Independent Component Analysis and Blind Source Separation (ICA'06)*, 2006, pp. 593–600.

[24] A. T. Cemgil, C. Févotte, and S. J. Godsill, "Variational and stochastic inference for Bayesian source separation," *Digital Signal Processing*, vol. 17, pp. 891–913, 2007.

[25] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[26] D. P. Wipf, J. A. Palmer, and B. D. Rao, "Perspectives on sparse Bayesian learning," in *Neural Inform. Process. Syst.*, 2004, vol. 16.

[27] E. G. Larrson and Y. Selen, "Linear regression with sparse parameter vector," *IEEE Trans. Signal Processing*, vol. 55, no. 2, pp. 451–460, Feb. 2007.

[28] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," in *IEEE Information Theory and Applications Workshop*, 2008, pp. 326–333.

[29] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Decoding real field codes by an iterative Expectation-Maximization (EM) algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'*, 2008.

[30] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," June 2009.

[31] C. Herzet and A. Drémeau, "Sparse representation algorithms based on mean-field approximations," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'*, 2010.

[32] C. Herzet and A. Drémeau, "Bayesian pursuit algorithms," in *EURASIP European Signal Processing Conference, EUSIPCO'*, 2010.

[33] A. Drémeau and C. Herzet, "Soft bayesian pursuit algorithm for sparse representations," in *IEEE International Workshop on Statistical Signal Processing 2011 (SSP'11)*, 2011.

[34] K. Qiu and A. Dogandzic, "ECME thresholding methods for sparse signal reconstruction," 2011.

[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, New York, USA, 1991.

[36] Dimitri P. Bertsekas and Dimitri P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, Sept. 1999.

[37] A. J. Miller, *Subset Selection in Regression*, Chapman & Hall/CRC, 2002.

[38] Christophe Couvreur and Yoram Bresler, "On the optimality of the backward greedy algorithm for the subset selection problem," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 797–808, Feb. 2000.

[39] M. A. Efroymson, *Multiple Analysis Regression*, vol. 1 of *Mathematical Methods for Digital Computers*, pp. 191–203, Wiley, New York, USA, 1960.

[40] Kenneth N. Berk, "Forward and backward stepping in variable selection," *Journal of Statistical Computation and Simulation*, vol. 10, no. 3-4, pp. 177–185, Apr. 1980.

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

[42] D. Haugland, "A bidirectional greedy heuristic for the subspace selection problem," vol. 4638 of *Lecture Notes in Computer Science*, pp. 162–176. Springer Berlin / Heidelberg, 2007.

[43] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Trans. Inform. Theory*, 2011.

[44] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[45] M. J. Beal and Z. Ghahramani, "The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures," *Bayesian Statistics*, vol. 7, 2003.

[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] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Bayesian pursuit algorithm for sparse representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'*, 2009.

[48] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing Communications and Control*, Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, 1995.

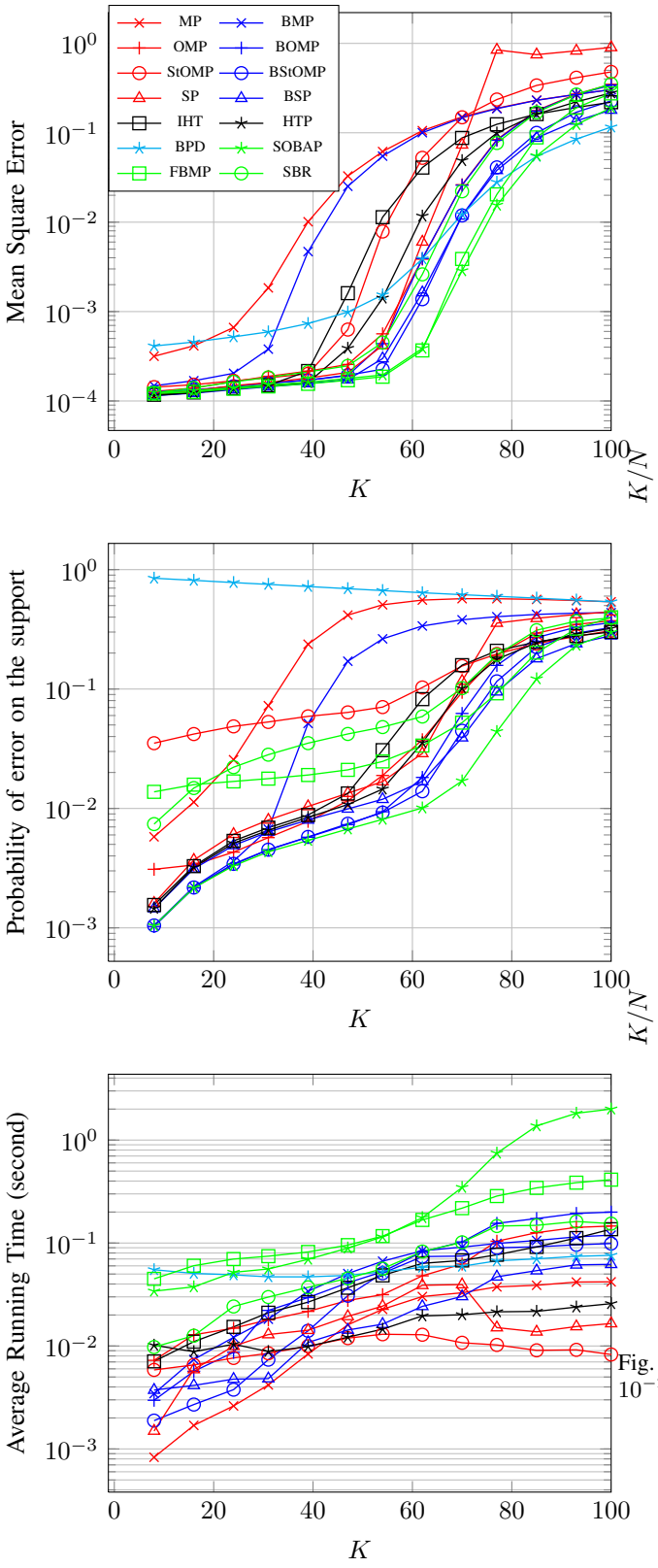

Fig. 1. MSE, probability of error on the support and average running time versus the number of non-zero coefficients K in the sparse vector.

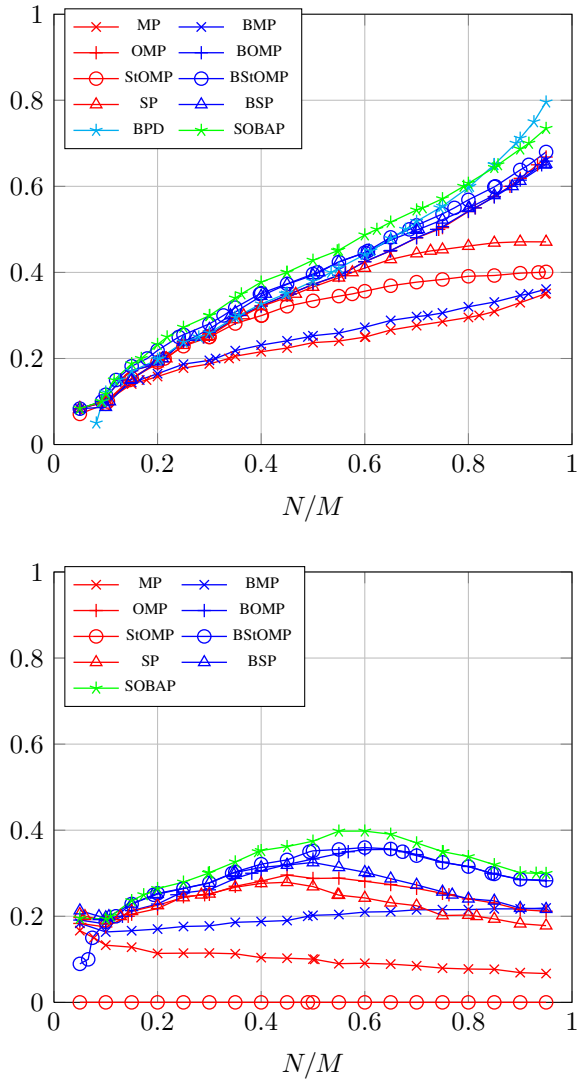

Fig. 2. Phase transition curves for MSE=10^{-2} (top) and $P_e = 10^{-2}$ (bottom).

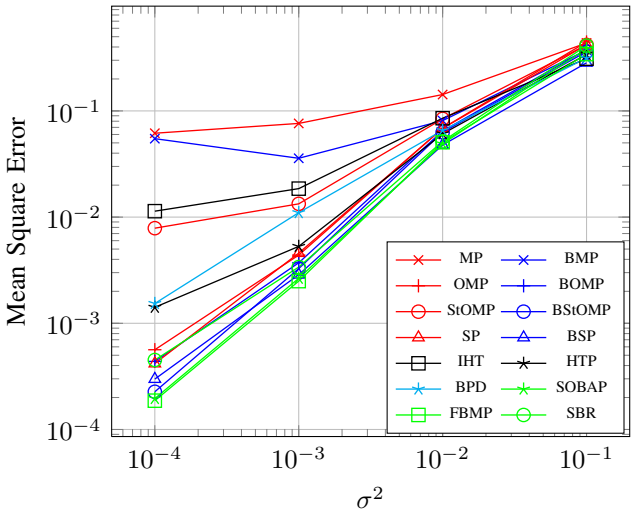

Fig. 3. MSE versus σ^2 for $K = 54$, $N = 154$ and $M = 256$.

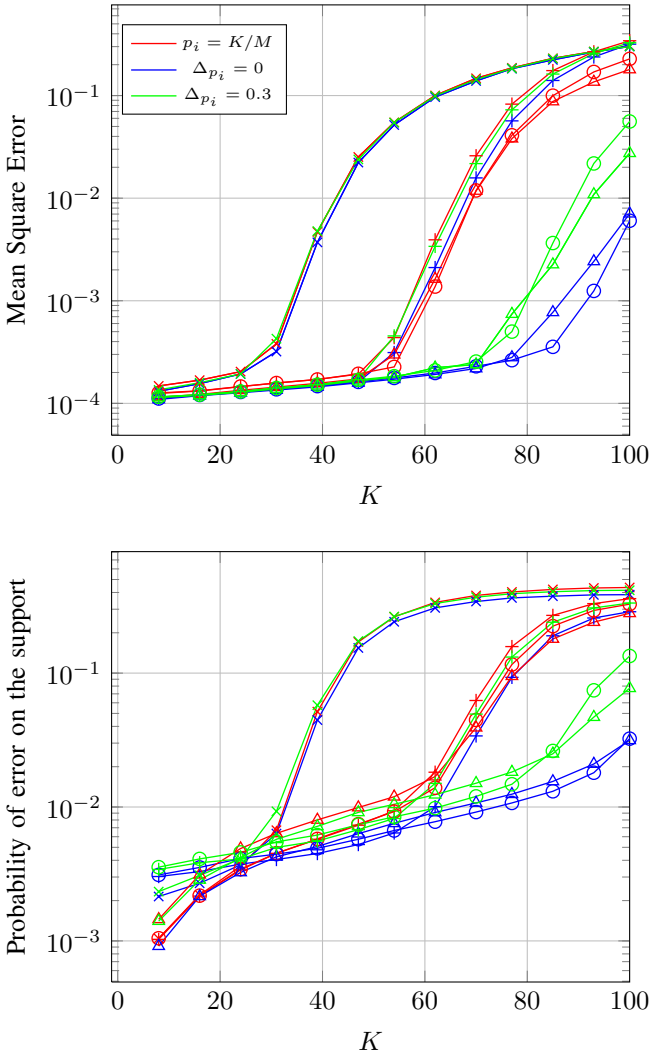

Fig. 4. MSE and probability of error on the support versus the number of non-zero coefficients K for BMP ($\times$), BOMP (+), BStOMP ($\circ$) and BSP ($\triangle$).

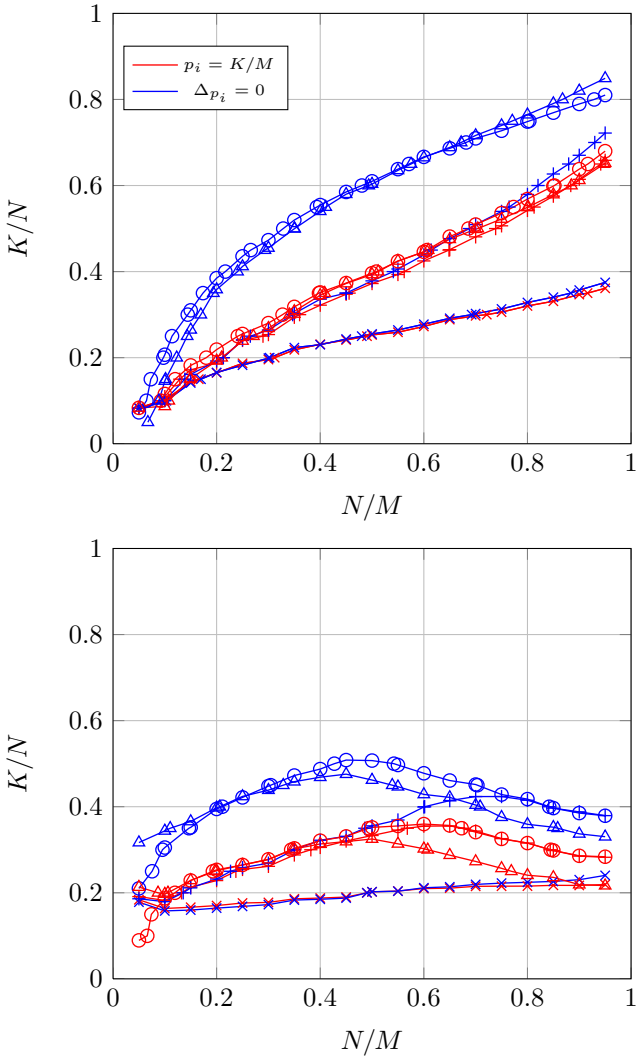

Fig. 5. Phase transition curves for MSE=10^{-2} (top) and $P_e = 10^{-2}$ (bottom) for BMP ($\times$), BOMP (+), BStOMP ($\circ$) and BSP ($\triangle$).