BAYESIAN PURSUIT ALGORITHMS

Cédric Herzet and Angélique Drémeau

INRIA Centre Rennes - Bretagne Atlantique,
Campus universitaire de Beaulieu, 35000 Rennes, France
phone: + (33) 2 99 84 73 50, fax: + (33) 2 99 84 71 71, email: {cedric.herzet, angelique.dremeau}@irisa.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 and explain several well-known pursuit algorithms (*e.g.*, MP, OMP, StOMP, CoSaMP, SP) as particular cases of the proposed Bayesian formulation.

1. INTRODUCTION

Sparse representations (SR) aim at describing a signal as the combination of a small number of atoms chosen from an overcomplete dictionary. More precisely, let $\mathbf{y}\in\mathbb{R}^N$ be an observed signal and $\mathbf{D}\in\mathbb{R}^{N\times M}$ a rank-N matrix whose columns are normalized to 1. Then, one standard formulation of the sparse representation problem writes

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 \leq \varepsilon, \tag{1}$$

or, in its Lagrangian version,

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

where $\|\cdot\|_p$ denotes the l_p-norm[1] and $\varepsilon,\lambda>0$ are parameters specifying the trade-off between sparsity and distortion.

Finding the exact solution of (1)-(2) is usually an intractable problem. Instead, suboptimal algorithms have been devised in the literature. We can roughly divide the existing algorithms into 3 main families: *i) the pursuit algorithms*, like matching pursuit (MP) [1], orthogonal matching pursuit (OMP) [2], stagewise OMP (StOMP) [3], subspace pursuit (SP) [4] or compressive sampling matching pursuit (CoSaMP) [5] build up the sparse vector $\mathbf{x}$ by making a succession of greedy decisions; *ii) the algorithms based on a problem relaxation*, like basis pursuit (BP) [6], FOCUSS [7] or SL0 [8], approximate (1)-(2) by relaxed problems which can be solved efficiently by standard optimization procedures; *iii) the Bayesian algorithms* express the SR problem as the solution of Bayesian inference problem and apply statistical tools to solve it. Examples of such algorithms include the relevant vector machine (RVM) algorithms [9], the sum-product and the expectation-maximization SR algorithms proposed in [10] and [11] respectively.

Whereas the connection between the pursuit/relaxation-based algorithms and the standard problem (1)-(2) is usually clear, it is not the case for the Bayesian algorithms available in the literature. In this paper we show that, under some conditions, the standard sparse representation problem (2) can be considered as a limit case of a maximum a posteriori (MAP) problem involving Bernoulli-Gaussian (BG) variables. This interpretation gives new insights into several existing pursuit algorithms and paves the way for the design of new ones.

[1] $\|\mathbf{x}\|_0$ denotes the number of non-zero elements in $\mathbf{x}$.

Thus, we exploit the equivalence between the standard and the BG MAP problems to derive novel Bayesian pursuit algorithms. The proposed algorithms generalize standard pursuit procedures in several aspects: *i)* they can exploit prior information about the atom occurrence and/or the amplitude of active coefficients; *ii)* unlike most of the existing pursuit procedures, they naturally implement the process of atom *deselection*; *iii)* the estimation of model parameters (noise variance, etc) can be nicely included within the considered Bayesian framework.

The rest of the paper is organized as follows. In section 2, we present a BG probabilistic framework modeling sparse processes and establish a connection between the standard problem and a maximum a posteriori (MAP) problem involving this model. In section 3 , we briefly review some well-known standard pursuit procedures. Section 4 is devoted to the derivation of Bayesian pursuit algorithms. Simulation results showing the good performance of the proposed approach are exposed in section 5.

2. A BAYESIAN FORMULATION OF THE STANDARD SR PROBLEM

Let $\mathbf{s}\in\{0,1\}^M$ be a vector defining the *support* of the sparse representation, *i.e.*, the subset of columns of $\mathbf{D}$ used to generate $\mathbf{y}$. Without loss of generality, we will 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:

$$\mathbf{y} = \sum_{i=1}^{M} s_i x_i \mathbf{d}_i + \mathbf{w}, \tag{3}$$

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

$$p(\mathbf{y}|\mathbf{x},\mathbf{s}) = \mathcal{N}(\mathbf{D}_\mathbf{s}\mathbf{x}_\mathbf{s}, \sigma_w^2\mathbf{I}_N), \tag{4}$$

where $\mathbf{I}_N$ is the $N\times N$-identity matrix and $\mathbf{D}_\mathbf{s}$ (resp. $\mathbf{x}_\mathbf{s}$) is a matrix (resp. vector) made up of the $\mathbf{d}_i$'s (resp. x_i's) such that $s_i=1$. We suppose that $\mathbf{x}$ and $\mathbf{s}$ obey the following probabilistic model:

$$p(\mathbf{x}) = \prod_{i=1}^{M} p(x_i), \qquad p(\mathbf{s}) = \prod_{i=1}^{M} p(s_i), \tag{5}$$

where

$$p(x_i) = \mathcal{N}(0,\sigma_x^2), \tag{6}$$
$$p(s_i) = \text{Ber}(p_i), \tag{7}$$

and $\text{Ber}(p_i)$ denotes a Bernoulli distribution with parameter p_i.

It is important to note that (4)-(7) 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 (4)-(7) is well-suited to modeling 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*[2] be non-zero, *i.e.*, the

[2] In an information-theoretic sense, *i.e.*, according to model (4)-(7), a realization of $\mathbf{s}$ with a few non-zero components will be observed with probability almost 1.

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 $\mathbf{D}$.

Model (4)-(7) (or variants thereof) has already been used in many Bayesian algorithms available in the literature, see *e.g.*, [10, 11, 12, 13]. However, to the best of our knowledge, no connection with the standard problem (2) has been made to date. The following result gives a Bayesian interpretation of standard problem (2) as a limit case of a MAP estimation problem involving the BG model defined in (4)-(7):

Theorem 1: Consider the following MAP estimation problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{s}}) = \underset{(\mathbf{x},\mathbf{s})}{\arg\max} \; \log p(\mathbf{y}, \mathbf{x}, \mathbf{s}), \tag{8}$$

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 (4)-(7).

If,

i) $\|\mathbf{D}_{\mathbf{s}}^{\dagger}\mathbf{y}\|_0 = \|\mathbf{s}\|_0$ with probability 1, $\forall \mathbf{s} \in \{0,1\}^M$, where $\mathbf{D}^{\dagger}_{\mathbf{s}^{(n)}}$ denotes the Moore-Penrose pseudo inverse of $\mathbf{D}_{\mathbf{s}^{(n)}}$.

ii) $\sigma_x^2 \to \infty$, $p_i = p \; \forall \; i$ and $\lambda = 2\sigma_w^2 \log(\frac{1-p}{p})$,

then, with probability 1,

$$\mathbf{x}^{\star} = \hat{\mathbf{x}}, \tag{9}$$

i.e., the solution of the BG MAP problem (8) is equal to the solution of standard SR problem (2). □

A proof of this result can be found in the appendix. Condition *i)* is only technical and ensures to discard some "pathological" cases. It is satisfied in most practical settings encountered in practice. In particular, this condition is verified as soon as $\mathbf{y}$ is a continuous random variable on $\mathbb{R}^N$.

The result established in Theorem 1 recasts the standard sparse representation problem (2) into a more general Bayesian framework. In particular, it reveals the statistical assumptions which are implicitly made when considering problem (2). It is interesting to note that the Bayesian formulation allows for more degrees of freedom than (2). For example, any prior information about the atom occurrence (p_i's) or the amplitude of the non-zero coefficients (σ_x^2) can explicitly be taken into account. The particular case $\sigma_x^2 = \infty$ corresponds to a non-informative prior $p(\mathbf{x})$.

Not surprisingly, the BG MAP formulation (8) does not offer any advantage in terms of complexity with respect to (2), *i.e.*, it is NP-hard. The practical computation of solutions of (8) requires therefore to resort to approximated (but practical) algorithms. In the rest of this paper, we will propose several greedy algorithms dealing with this task. Due to the equivalence (9), the proposed greedy procedures will share some similarities with standard pursuit algorithms.

3. STANDARD PURSUIT ALGORITHMS

In this section, we briefly recall the process of standard pursuit algorithms. In particular, we dwell upon four of the most popular, namely MP, OMP, StOMP and CoSaMP/SP[3].

Standard pursuit algorithms iterate between 2 main steps:

Support update: the algorithm updates the support of the sparse representation, *i.e.*, makes a guess about the columns (or *atoms*) of the dictionary which have been used to generate $\mathbf{y}$.

Coefficient update: the estimate of $\mathbf{x}$ is refined by taking into account the latest decision about the support.

[3]CoSaMP and SP are two slightly different versions of the same algorithm (see [4] and [5]).

MP, OMP, StOMP and CoSaMP/SP basically differ in the way they implement these two steps.

The MP algorithm performs iteratively the following steps:

$$\hat{s}_j^{(n)} = \begin{cases} 1 & \text{if } j = \underset{i}{\arg\max} \; \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2, \\ \hat{s}_j^{(n-1)} & \text{otherwise,} \end{cases} \tag{10}$$

$$\hat{x}_j^{(n)} = \begin{cases} \hat{x}_j^{(n-1)} + \langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle & \text{if } j = \underset{i}{\arg\max} \; \langle \mathbf{r}^{(n)}, \mathbf{d}_i \rangle^2, \\ \hat{x}_j^{(n-1)} & \text{otherwise,} \end{cases} \tag{11}$$

where $\langle \mathbf{u}, \mathbf{v} \rangle \triangleq \mathbf{u}^T \mathbf{v}$ denotes vector inner product and $\mathbf{r}^{(n)}$ is the current residual:

$$\mathbf{r}^{(n)} \triangleq \mathbf{y} - \sum_j \hat{x}_j^{(n-1)} \mathbf{d}_j. \tag{12}$$

At each iteration, MP adds *at most* one single atom to the support based on the amplitude of its projection with the residual. It can be seen that this support update strategy maximizes the decrease of the residual norm at each iteration.

OMP performs the same support update as MP but computes the coefficient estimate in a different way. Let $\hat{\mathbf{s}}^{(n)}$ define the support estimate at iteration n. Then, OMP computes an estimate of the *non-zero* coefficients as follows:

$$\hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n)}} = \mathbf{D}^{\dagger}_{\hat{\mathbf{s}}^{(n)}} \mathbf{y} = \left(\mathbf{D}^T_{\hat{\mathbf{s}}^{(n)}} \mathbf{D}_{\hat{\mathbf{s}}^{(n)}} \right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}^{(n)}} \mathbf{y}, \tag{13}$$

where $\mathbf{D}^{\dagger}_{\hat{\mathbf{s}}^{(n)}}$ represents the Moore-Penrose pseudo inverse of $\mathbf{D}_{\hat{\mathbf{s}}^{(n)}}$.

StOMP is a modified version of OMP which allows for the selection of several new atoms at each iteration. The choice of the atoms added to the support estimate $\hat{\mathbf{s}}^{(n)}$ is made by a threshold decision on $\langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle^2$:

$$\hat{s}_j^{(n)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle^2 > T^{(n)}, \\ \hat{s}_j^{(n-1)} & \text{otherwise,} \end{cases} \tag{14}$$

where $T^{(n)}$ is a threshold depending on the iteration number. In [3], the authors proposed two different approaches to tune the value of the threshold $T^{(n)}$ according to some criterion.

Common to MP, OMP and StOMP is the fact that atom *deselection* is not possible: once a column of $\mathbf{D}$ has been added to the support, it can never (explicitly) be removed. CoSaMP and SP provide a solution to this problem. These procedures rely on the following support-update rule:

$$\hat{\mathbf{s}}^{(n)} = \underset{\mathbf{s}}{\arg\max} \left\{ \sum_j s_j \, |\tilde{x}_j^{(n)}| \right\} \text{subject to } \|\mathbf{s}\|_0 = K, \tag{15}$$

where K denotes the number of atoms used to generate $\mathbf{y}$ and $\tilde{\mathbf{x}}^{(n)}$ is a trial coefficient estimate computed from (13) and using the following trial support estimate

$$\tilde{\mathbf{s}}^{(n)} = \underset{\mathbf{s}}{\arg\max} \left\{ \sum_j s_j \langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle^2 \right\} \text{subject to } \|\mathbf{s}\|_0 = P \quad \text{and } s_i = 1 \; \forall i \in \mathscr{I} \; , \tag{16}$$

with $\mathscr{I} = \{i \in \{1, \ldots, M\} | \hat{s}_i^{(n-1)} = 1\}$ and $P > K$. Clearly, updates (15)-(16) allow for the deselection of atoms throughout the iterative process. Note however, that CoSaMP and SP require the knowledge of the number of non-zero coefficients K.

4. BAYESIAN PURSUIT ALGORITHMS

In this section, we derive pursuit algorithms from the Bayesian framework described in section 2.

As previously mentioned, we will see that these algorithms turn out to be extensions of standard pursuit algorithms (see section 3). They offer in particular highest flexibility and precision in the computation of the support and coefficient estimates:

- The prior information about the occurrence of each atom in the sparse decomposition, *i.e.*, p_i's, can explicitly be taken into account into the estimation process.
- The problem of column deselection is naturally solved.
- The Bayesian framework allows for model parameter estimation. In particular, we will see that the estimation of the noise variance throughout the iterations plays a crucial role in the algorithm performance.

The proposed algorithms are tractable procedures searching for the solution of (8) by iterative greedy maximization of $\log p(\mathbf{y},\mathbf{x},\mathbf{s})$. We describe hereafter four different greedy implementation of (8).

4.1 Bayesian Matching Pursuit (BMP)

As mentioned in section 3, MP updates at each iteration the coefficient leading to the maximum decrease of the residual norm. A similar approach can be followed within the Bayesian framework considered here: the BMP algorithm can be defined so that the couple (s_j,x_j) updated at each iteration locally maximizes the increase of $\log p(\mathbf{y},\mathbf{x},\mathbf{s})$.

In order to properly describe this procedure, let us first define

$$\rho^{(n)}(s_j,\hat{\mathbf{x}}^{(n-1)}) \triangleq \max_{x_j}\left\{\log\frac{p(\mathbf{y},\hat{\mathbf{x}}_j^{(n-1)},\hat{\mathbf{s}}_j^{(n-1)})}{p(\mathbf{y},\hat{\mathbf{x}}^{(n-1)},\hat{\mathbf{s}}^{(n-1)})}\right\}, \qquad (17)$$

where $\hat{\mathbf{x}}_j^{(n)}$ (resp. $\hat{\mathbf{s}}_j^{(n)}$) is a vector equal to $\hat{\mathbf{x}}^{(n)}$ (resp. $\hat{\mathbf{s}}^{(n)}$) but for the jth component which is free to vary. Therefore, $\rho^{(n)}(s_j,\hat{\mathbf{x}}^{(n-1)})$ represents the variation of the goal function when optimized over x_j while all other variables are kept fixed. Note that this variation is a function of the value assigned to $s_j \in \{0,1\}$.

We define the Bayesian MP (BMP) algorithm by the following recursions:

- **BMP support update:**

$$\hat{s}_j^{(n)} = \begin{cases} \tilde{s}_j^{(n)} & \text{if } j = \arg\max_i \rho^{(n)}(\tilde{s}_i^{(n)},\hat{\mathbf{x}}^{(n-1)}), \\ \hat{s}_j^{(n-1)} & \text{otherwise.}\end{cases} \qquad (18)$$

where

$$\tilde{s}_j^{(n)} \triangleq \arg\max_{s_j\in\{0,1\}} \rho^{(n)}(s_j,\hat{\mathbf{x}}^{(n-1)}),$$
$$= \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_j^{(n-1)}\mathbf{d}_j,\mathbf{d}_j\rangle^2 > T_j, \\ 0 & \text{otherwise.}\end{cases} \qquad (19)$$

with

$$T_j \triangleq 2\sigma_w^2\frac{\sigma_x^2+\sigma_w^2}{\sigma_x^2}\log\left(\frac{1-p_j}{p_j}\right). \qquad (20)$$

- **BMP coefficient update:**

$$\hat{x}_j^{(n)} = \begin{cases} \tilde{x}_j^{(n)} & \text{if } j = \arg\max_i \rho^{(n)}(\tilde{s}_i^{(n)},\hat{\mathbf{x}}^{(n-1)}), \\ \hat{x}_j^{(n-1)} & \text{otherwise.}\end{cases} \qquad (21)$$

where

$$\tilde{x}_j^{(n)} = \arg\max_{x_j} \log p(\mathbf{y},\hat{\mathbf{x}}_j^{(n-1)},\hat{\mathbf{s}}^{(n)}),$$
$$= \hat{s}_j^{(n)}\left(\hat{x}_j^{(n-1)} + \langle \mathbf{r}^{(n)},\mathbf{d}_j\rangle\right)\frac{\sigma_x^2}{\sigma_x^2+\sigma_w^2}. \qquad (22)$$

We can make the following comments about these recursions:

- Since the procedure described in (18)-(22) corresponds to a sequential maximization of the upper-bounded function $\log p(\mathbf{y},\mathbf{x},\mathbf{s})$, the convergence to a fixed point, say $(\hat{\mathbf{x}}^{(\infty)},\hat{\mathbf{s}}^{(\infty)})$, is ensured. Moreover, the fixed points must be "local" maxima[4] of $\log p(\mathbf{y},\mathbf{x},\mathbf{s})$.
- The algorithm complexity is similar to MP: the most expensive operation is the maximization in (18) which scales as $\mathcal{O}(M)$ (we omit the details here due to space limitation).
- $\tilde{s}_j^{(n)}$ is the locally-optimal decision about s_j, *i.e.*, the decision maximizing the increase of the goal function given the current estimate. The value of $\tilde{s}_j^{(n)}$ is based on the comparison of a signal energy in the direction of $\mathbf{d}_j$ to a threshold T_j (see (19)). This threshold depends on the probability of occurrence of each atoms, p_j : the larger p_j the smaller T_j and the more likely is the column to be selected in the sparse representation. Note that if $\tilde{s}_j^{(n)} = 0$ whereas $\hat{s}_j^{(n-1)} = 1$, the locally-optimal decision consists in *removing* column $\mathbf{d}_j$ from the support. As mentioned earlier, the BMP algorithm therefore naturally implements the process of deselecting some of the columns of the current support.
- The update of the coefficient amplitude (see (22)) is made by taking into account some prior information about the distribution of $\mathbf{x}$, *i.e.*, σ_x^2. Note that if $\hat{s}_j^{(n)} = 1$ and $\sigma_x^2 \to \infty$, (22) becomes

$$\tilde{x}_j^{(n)} = \hat{x}_j^{(n-1)} + \langle \mathbf{r}^{(n)},\mathbf{d}_j\rangle. \qquad (23)$$

i.e., we recover the MP coefficient update (11).

In section 2, we emphasized that the joint BG MAP problem (8) and the standard SR problem (2) are equivalent when $\sigma_x^2 \to \infty$ and $p_i = p\ \forall\ i$. These conditions are not sufficient to ensure the equivalence between BMP and MP algorithms[5] because of the atom deselection, allowed by BMP but impossible in the MP procedure. Withdrawing this opportunity (by forcing $\tilde{s}_j^{(n)} = 1\ \forall\ j$), *i.e.*, only considering the addition (but never the removal) of new atoms in the support, one recovers standard MP implementation. The standard MP algorithm can therefore be regarded as a particular case of the Bayesian pursuit algorithm presented in this section.

4.2 Bayesian Orthogonal Matching Pursuit (BOMP)

We now consider the implementation of the Bayesian orthogonal matching pursuit by modifying the coefficient-update step of the BMP algorithm. In particular, BOMP computes the estimate of $\mathbf{x}$ as follows:

$$\hat{\mathbf{x}}^{(n)} = \arg\max_{\mathbf{x}} \log p(\mathbf{y},\mathbf{x},\hat{\mathbf{s}}^{(n)}). \qquad (24)$$

Solving this problem, we obtain that the $\hat{x}_j^{(n)}$'s such that $\hat{s}_j^{(n)} = 1$ are given by

$$\hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n)}}^{(n)} = \left(\mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T\mathbf{D}_{\hat{\mathbf{s}}^{(n)}} + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_{\|\hat{\mathbf{s}}^{(n)}\|_0}\right)^{-1}\mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T\mathbf{y}, \qquad (25)$$

[4] Concerning $\mathbf{s}$ which takes on values in a finite set, the local optimality has to be understood as follows: there is no modification of *one* single component of $\hat{\mathbf{s}}^{(\infty)}$ that leads to an increase of the goal function.

[5] This can readily be shown by using $\sigma_x^2 \to \infty$ and $p_i = p\ \forall\ i$ in recursions (18)-(22). We omit however the details here due to space limitation.

and $\hat{x}_j^{(n)} = 0$ otherwise. Observe that, like BMP, BOMP updates non-zero coefficients by taking into account the prior information about the coefficient amplitude, σ_x^2.

The update of the support remains unchanged with respect to BMP. Hence, like BMP, BOMP also implements atom deselection. For this reason, similar to the one mentioned for the BMP/MP equivalence, BOMP does not reduce to OMP when $\sigma_x^2 \to \infty$ and $p_i = p \; \forall \; i$. Finally, from the same reasoning as for BMP, it can be seen that BOMP converges to local maxima of $\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$.

4.3 Bayesian Stagewise Orthogonal Matching Pursuit (BStOMP)

BStOMP is a modified version of BOMP where several entries of the support vector $\mathbf{s}$ can be changed at each iteration. We propose the following approach:

$$\hat{s}_j^{(n)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)} + \hat{x}_j^{(n-1)} \mathbf{d}_j, \mathbf{d}_j \rangle^2 > T_j, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

where T_j is defined in (20). Note that if the jth atom was not selected at iteration $n-1$, *i.e.*, $(\hat{x}_j^{(n-1)}, \hat{s}_j^{(n-1)}) = (0,0)$, (26) becomes

$$\hat{s}_j^{(n)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n)}, \mathbf{d}_j \rangle^2 > T_j, \\ \hat{s}_j^{(n-1)} & \text{otherwise.} \end{cases} \tag{27}$$

In such a case, the support update rules of StOMP and BStOMP are therefore similar. However, in the general case (26), BStOMP allows for the deselection of atoms.

Another crucial difference between StOMP and BStOMP is the definition of the threshold T_j. 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 [3], it requires therefore no additional hypothesis and/or design criterion.

Finally, let us mention that the performance of BStOMP can be greatly improved by including the estimation of the noise variance σ_w^2 in the iterative process. As mentioned earlier, the estimation of model parameters is naturally included in the Bayesian framework considered in this paper. In particular, the maximum-likelihood (ML) estimate of σ_w^2 writes

$$\begin{aligned} (\hat{\sigma_w^2})^{(n)} &= \arg\max_{\sigma_w^2} \log p(\mathbf{y}, \hat{\mathbf{x}}^{(n-1)}, \hat{\mathbf{s}}^{(n-1)}), \\ &= N^{-1} \|\mathbf{y} - \mathbf{D}\hat{\mathbf{x}}^{(n-1)}\|_2^2 = N^{-1} \|\mathbf{r}^{(n)}\|_2^2. \end{aligned} \tag{28}$$

Plugging this expression into (20), we obtain:

$$T_j^{(n)} \triangleq 2 \frac{\|\mathbf{r}^{(n)}\|_2^2}{N} \frac{\sigma_x^2 + N^{-1} \|\mathbf{r}^{(n)}\|_2^2}{\sigma_x^2} \log\left(\frac{1-p_j}{p_j}\right). \tag{29}$$

The threshold therefore becomes a function of the iteration number. Note that, when $\sigma_x^2 \to \infty$, $T_j^{(n)}$ has the following expression:

$$T_j^{(n)} \overset{\sigma_x^2 \to \infty}{=} 2 \frac{\|\mathbf{r}^{(n)}\|_2^2}{N} \log\left(\frac{1-p_j}{p_j}\right). \tag{30}$$

$T_j^{(n)}$ is then proportional to the residual energy; the factor of proportionality depends on the probability of occurrence of each atom.

4.4 Bayesian Subspace Pursuit (BSP)

We finally propose a Bayesian pursuit algorithm having some flavor of CoSaMP/SP. We will refer to this algorithm as Bayesian subspace pursuit (BSP) algorithm.

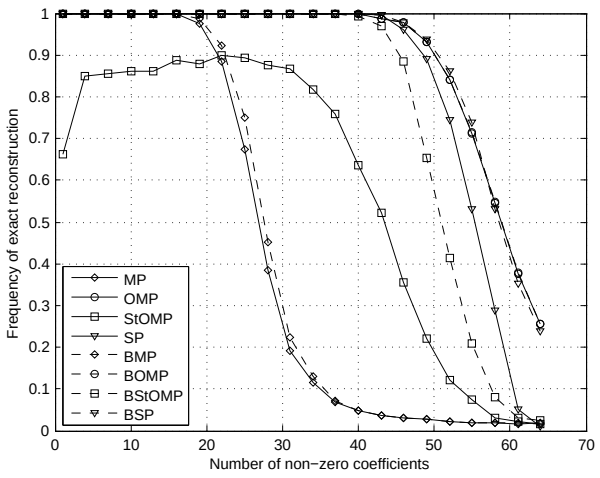

Figure 1: Frequency of exact reconstruction versus number of non-zero coefficients; $N = 128$, $M = 256$, $\sigma_w^2 = 10^{-5}$, $\sigma_x^2 = 10$.

We define the support update performed by BSP as follows:

$$\hat{\mathbf{s}}^{(n)} = \arg\max_{\mathbf{s}} \left\{ \sum_j \rho^{(n)}(s_j, \tilde{\mathbf{x}}^{(n)}) \right\} \text{subject to } \|\mathbf{s}\|_0 = K, \tag{31}$$

where $\tilde{\mathbf{x}}^{(n)}$ is a trial coefficient estimate computed from (24) by using $\tilde{\mathbf{s}}^{(n)}$ as support estimate:

$$\tilde{\mathbf{s}}^{(n)} = \arg\max_{\mathbf{s}} \left\{ \sum_j \rho^{(n)}(s_j, \hat{\mathbf{x}}^{(n-1)}) \right\}. \tag{32}$$

A new coefficient estimate $\hat{\mathbf{x}}^{(n)}$ is finally computed from (24).

It is interesting to note that, unlike CoSaMP/SP, BSP imposes no constraint on the number of non-zero elements in $\tilde{\mathbf{s}}^{(n)}$. In particular, $\|\tilde{\mathbf{s}}^{(n)}\|_0$ can be larger or lower than K. In fact, $\tilde{\mathbf{s}}^{(n)}$ is computed by making the best local decision for each atom of the dictionary. This is equivalent to the support update rule implemented by StOMP in (26). The support estimate $\hat{\mathbf{s}}^{(n)}$ is then computed by only keeping in the support the K columns having the largest components $\tilde{x}_j^{(n)}$.

5. SIMULATION RESULTS

In this section, we study the performance of the proposed SR algorithms by extensive computer simulations. We follow the same methodology as in [4] to assess the performance of the SR algorithms: we calculate the empirical frequency of correct reconstruction versus the number of non-zero coefficients in $\mathbf{x}$, say K. We assume that a vector has been correctly reconstructed when the amplitude of the error reconstruction on each non-zero coefficient is lower than 10^{-4}.

Fig. 1 illustrates the performance achieved by BMP, BOMP, BStOMP, BSP and MP, OMP, StOMP, SP. We use the following parameters for the generation of these curves: $N = 128$, $M = 256$, $\sigma_w^2 = 10^{-5}$. For the sake of fair comparison with standard pursuit algorithms, we consider the case where all the atoms have the same probability of occurrence, *i.e.*, $p_j = K/M \; \forall j$. The data is therefore generated as follows. The positions of the non-zero coefficients are first drawn uniformly at random. Then, the amplitude of the non-zero coefficients is generated from a zero-mean Gaussian with variance $\sigma_x^2 = 10$. The elements of the dictionary are *i.i.d* realizations of a zero-mean Gaussian distribution with variance N^{-1}. For each

point of simulation, we run 400 trials. In order not to favor our methods with any additional prior information, we use $\sigma_x^2 = 1000$ in the proposed Bayesian algorithms.

MP and OMP are run until the l_2-norm of the residual drops below $\sqrt{N\sigma_w^2}$. The Bayesian pursuit algorithms iterate as long as $\log p(\mathbf{y}, \hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)}) > \log p(\mathbf{y}, \hat{\mathbf{x}}^{(n-1)}, \hat{\mathbf{s}}^{(n-1)})$. We use the *SparseLab* implementation of StOMP available at *http://sparselab.stanford.edu/* and SP implementation available at *http://igorcarron.googlepages.com/cscodes*. StOMP is used with the (so-called) CFDR threshold criterion. BStOMP and BSP consider thresholding based on noise variance estimates (29).

We observe that the proposed Bayesian algorithms improve the performance upon their standard version. The gain in performance depends on the algorithms. On the one hand BMP leads to a small improvement whereas the performance of OMP and BOMP overlaps. We observe however that BOMP decreases the computational time by a factor between 5 and 10 with respect to OMP. This is a consequence of the atom deselection process which efficiently reduces the size of the support when required. On the other hand BStOMP and BSP exhibit a clear superiority with respect to StOMP and SP. Note that BSP achieves the same performance as BOMP/OMP but with a computational time similar to SP, *i.e.*, roughly 50 times smaller than OMP.

6. CONCLUSION

In this paper, we addressed the sparse representation (SR) problem within a general Bayesian framework. We first showed the equivalence between the standard SR formulation and a maximum a posteriori (MAP) problem involving Bernoulli-Gaussian variables. We exploited this result to give a Bayesian generalization of well-known standard pursuit algorithms. We emphasized theoretical advantages of the proposed algorithms, like atom deselection and parameter estimation, and confirmed them by some practical experiments.

7. APPENDIX: PROOF OF THEOREM 1

Let $f(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0$ and $\mathbf{x}^\star(\mathbf{s})$ be the solution of

$$\mathbf{x}^\star(\mathbf{s}) = \arg\min_{\mathbf{x}} f(\mathbf{x}) \;\text{ s.t. } x_i = 0 \;\text{ if } s_i = 0. \tag{33}$$

$\mathbf{x}^\star(\mathbf{s})$ is therefore the optimal solution of the standard problem if the position of the non-zero coefficients is specified. Note that the notation $\mathbf{x}^\star(\mathbf{s})$ is somehow misleading since the solution of the "arg min"-problem in (33) is non-unique if $\mathbf{D_s}$ is not full-rank. For the sake of conciseness we will restrict the demonstration hereafter to the case where $\|\mathbf{s}\|_0 \leq N$ and every subset of $L \leq N$ columns of $\mathbf{D}$ are linearly independent. This implies that $\mathbf{D_s}$ is full-rank $\forall \mathbf{s}$. The general case is similar although slightly more involved.

Clearly, the solution of (2) can thus be reformulated as

$$\mathbf{x}^\star = \mathbf{x}^\star(\mathbf{s}^\star) \;\text{ with } \mathbf{s}^\star = \arg\min_{\mathbf{s}\in\{0,1\}^M} f(\mathbf{x}^\star(\mathbf{s})). \tag{34}$$

Similarly, let $g(\mathbf{x}) \triangleq -\log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$ and $\hat{\mathbf{x}}(\mathbf{s})$ be the solution of $\hat{\mathbf{x}}(\mathbf{s}) = \arg\max_{\mathbf{x}} \log p(\mathbf{y}, \mathbf{x}, \mathbf{s})$. Problem (8) can then be reformulated as:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}(\hat{\mathbf{s}}) \;\text{ with } \hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\{0,1\}^M} g(\hat{\mathbf{x}}(\mathbf{s})). \tag{35}$$

Theorem 1 can therefore be proved by showing that $\hat{\mathbf{x}}(\mathbf{s}) = \mathbf{x}^\star(\mathbf{s})$ and $g(\hat{\mathbf{x}}(\mathbf{s})) = f(\mathbf{x}^\star(\mathbf{s}))\ \forall \mathbf{s}$ under the considered hypotheses.

Without loss of generality, we assume that the first k components of $\mathbf{s}$ are non-zero. If $\mathbf{D_s}$ denotes the matrix made up of the first k columns of $\mathbf{D}$ and $\mathbf{D_s^\dagger}$ its Moore-Penrose pseudo-inverse, we then have

$$x_i^\star(\mathbf{s}) = \begin{cases} \left(\mathbf{D_s^\dagger}\mathbf{y}\right)_i & i \in \{1,\ldots,k\}, \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, the solution of (35) writes

$$\hat{x}_i(\mathbf{s}) = \begin{cases} \left(\left(\mathbf{D_s}^T\mathbf{D_s} + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I}_k\right)^{-1}\mathbf{D_s}^T\mathbf{y}\right)_i & i \in \{1,\ldots,k\}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $\lim_{\sigma_x^2\to\infty} \hat{\mathbf{x}}(\mathbf{s}) = \mathbf{x}^\star(\mathbf{s})$. Using this result and taking (4)-(7) into account, we have

$$\lim_{\sigma_x^2\to\infty} g(\hat{\mathbf{x}}(\mathbf{s})) = \frac{\|\mathbf{y} - \mathbf{D}\mathbf{x}^\star(\mathbf{s})\|_2^2}{2\sigma_w^2} + \log p(\mathbf{s}) + \lim_{\sigma_x^2\to\infty} \frac{\sum_{i=1}^k (x_i^\star(\mathbf{s}))^2}{2\sigma_x^2}.$$

Note that the last term tends to zero when $\sigma_x^2 \to \infty$. Moreover, $p(\mathbf{s}) \propto \exp\{\|\mathbf{s}\|_0 \log(\frac{1-p}{p})\}$ if $p_i = p\ \forall i$. Now, we have by hypothesis that $\|\mathbf{x}^\star(\mathbf{s})\|_0 \triangleq \|\mathbf{D_s^\dagger}\mathbf{y}\|_0 = \|\mathbf{s}\|_0$ with probability one. Therefore, since $\lambda = 2\sigma_w^2 \log(\frac{1-p}{p})$, we have $g(\hat{\mathbf{x}}(\mathbf{s})) = f(\mathbf{x}^\star(\mathbf{s}))$ with probability one.

REFERENCES

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

[2] 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.

[3] D.L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," available at http://www-stat.stanford.edu/ donoho/reports.html, 2006.

[4] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," available at arXiv:0803.0811v3, January 2009.

[5] 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.

[6] 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.

[7] 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, March 1997.

[8] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed l^0 norm," *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 289–301, January 2009.

[9] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[10] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," available at arXiv:0812.4627v2, June 2009.

[11] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Sparse component analysis in presence of noise using EM-MAP," in *7th International Conference on Independent Component Anaysis and Signal Separation*, London, UK, 2007.

[12] 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.

[13] 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.