



Structured Bayesian Orthogonal Matching Pursuit

Angélique Drémeau, Cedric Herzet, Laurent Daudet

► **To cite this version:**

Angélique Drémeau, Cedric Herzet, Laurent Daudet. Structured Bayesian Orthogonal Matching Pursuit. IEEE ICASSP 2012, Mar 2012, Kyoto, Japan. 2012. <hal-00754995>

HAL Id: hal-00754995

<https://hal.inria.fr/hal-00754995>

Submitted on 20 Nov 2012

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

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

Structured Bayesian Orthogonal Matching Pursuit

Angélique Drémeau ^{a,b*}; Cédric Herzet ^c and Laurent Daudet ^{a†}

^a Institut Langevin, ESPCI ParisTech, Univ Paris Diderot,
CNRS UMR 7587, F-75005 Paris, France

^b Fondation Pierre-Gilles De Gennes pour la Recherche, 29 rue d'Ulm, F-75005 Paris, France

^c INRIA Centre Rennes - Bretagne Atlantique,
Campus universitaire de Beaulieu, F-35000 Rennes, France

November 20, 2012

Abstract

Taking advantage of the structures inherent in many sparse decompositions constitutes a promising research axis. In this paper, we address this problem from a Bayesian point of view. We exploit a Boltzmann machine, allowing to take a large variety of structures into account, and focus on the resolution of a joint maximum a posteriori problem. The proposed algorithm, called Structured Bayesian Orthogonal Matching Pursuit (SBOMP), is a structured extension of the Bayesian Orthogonal Matching Pursuit algorithm (BOMP) introduced in our previous work [1]. In numerical tests involving a recovery problem, SBOMP is shown to have good performance over a wide range of sparsity levels while keeping a reasonable computational complexity.

1 Introduction

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

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0, \quad (1)$$

*AD is currently working at Institut Télécom, Télécom ParisTech, CNRS-LTCl, F-75014 Paris, France.

†LD is on a joint affiliation with Université Paris Diderot - Paris 7 and the Institut Universitaire de France.

where $\|\mathbf{x}\|_0$ 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.

Finding the exact solution of (1) is an NP-hard problem: it generally requires a combinatorial search over the entire solution space. Therefore, heuristic (but tractable) algorithms have been devised to deal with this problem. As a well-known example, let us mention Orthogonal Matching Pursuit (OMP) [2].

More recently, the SR problem has been enhanced by the introduction of structural constraints on the support of the sparse representation: the non-zero components of \mathbf{x} can no longer be chosen arbitrarily but must obey some (deterministic or probabilistic) rules. This problem is often referred to as “*structured*” sparse representation. This new paradigm has been found to be relevant in many application domains and has recently sparked a surge of interest in algorithms coping with this problem. The procedures currently available in the literature can be classified according to the type of structures they exploit:

1) *Group sparsity*: in group-sparse signals, coefficients are either all non-zero or all zero within prespecified groups of atoms. One popular way to enforce group sparsity in sparse decompositions is the use of particular “mixed” norms combining ℓ_1 - and ℓ_2 -norms. The Group-LASSO and Block-OMP algorithms proposed in [3] and [4] follow this approach.

2) *Molecular sparsity*: molecular sparsity describes more complex structures, in the particular case where the atoms of the dictionary have a double indexation (*e.g.*, time-frequency atoms). Molecular sparsity can be exploited using a general definition of mixed norms. This approach has been followed by Kowalski and Torr sani in [5] for the derivation of the Elitist-LASSO algorithm.

3) *Chain and tree-structured sparsity*: Trees and chains are elementary structures arising in many signal-processing applications (*e.g.*, Markov chain, multi-resolution decomposition, etc.) The combination of a tree structure and sparse representations has been studied in [6]: the authors enforce a tree-structured sparsity by using a particular penalty term. In [7], F votte *et al.* consider a model promoting chain-structured sparsity via the use of a Markov-chain probabilistic model.

4) *Generic structured sparsity*: some more recent approaches do not focus on a specific type of structure but propose general models accounting for a wide set of structures. Most of these approaches are probabilistic. In particular, [8, 9] and [10] have recently emphasized the relevance of the Boltzmann machine as a general model for structured sparse representations.

In this paper, we address the problem of structured SR in a generic probabilistic model. We introduce a novel pursuit algorithm looking for a solution of a joint maximum a posteriori (MAP) problem and implementing the interconnections between the atoms of the support via a Boltzmann machine. The proposed algorithm can be seen as a generalization of the so-called Bayesian Orthogonal Matching Pursuit (BOMP) presented in [1]. Our numerical results show that the proposed procedure exhibits state-of-the-art performance in terms of reconstruction-complexity trade-off.

2 Probabilistic model

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 i th column of \mathbf{D} is (resp. is not) used to form \mathbf{y} . We assume that the columns of \mathbf{D} are normalized. Denoting by \mathbf{d}_i the i th column of \mathbf{D} , we then consider the following observation model:

$$\mathbf{y} = \sum_{i=1}^M s_i x_i \mathbf{d}_i + \mathbf{n}, \quad (2)$$

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

$$p(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \mathcal{N}(\mathbf{D}_\mathbf{s}\mathbf{x}_\mathbf{s}, \sigma^2\mathbf{I}_N), \quad (3)$$

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} obeys the following probabilistic model:

$$p(\mathbf{x}) = \prod_{i=1}^M p(x_i) \text{ where } p(x_i) = \mathcal{N}(0, \sigma_x^2), \quad (4)$$

and \mathbf{s} is distributed according to a Boltzmann machine of parameters \mathbf{b} and \mathbf{W} :

$$p(\mathbf{s}) \propto \exp(\mathbf{b}^T \mathbf{s} + \mathbf{s}^T \mathbf{W} \mathbf{s}), \quad (5)$$

where \propto denotes equality up to a normalization factor. \mathbf{W} is a symmetric matrix with zeros on the diagonal (elements of \mathbf{W} are denoted by w_{ij} , for the i th line and j th column).

Within model (3)-(5), the observation \mathbf{y} can thus be seen as the noisy combination of atoms specified by \mathbf{s} . The weights of the combination are realizations of Gaussian distributions whose variance is independent of the support \mathbf{s} .

The Boltzmann machine encompasses many well-known probabilistic models as particular cases. For example, the choice $\mathbf{W} = 0_{M \times M}$ leads to the Bernoulli model

$$p(\mathbf{s}) \propto \exp(\mathbf{b}^T \mathbf{s}) = \prod_i \exp(b_i s_i) = \prod_i \text{Ber}(p_i), \quad (6)$$

with $p_i = \frac{1}{1 + \exp(-b_i)}$. This model is well-known in the literature to address the unstructured SR problem (see *e.g.*, [1, 11]).

3 Joint MAP Estimation Problem

The probabilistic framework defined in section 2 allows us to tackle the SR problem from a Bayesian perspective. As long as (3)-(5) is the true generative

model for the observations \mathbf{y} , optimal estimators can be derived under different Bayesian criteria (mean square error, mean absolute error, etc). We focus hereafter on the computation of a solution under a joint maximum a posteriori (MAP) criterion

$$(\hat{\mathbf{x}}, \hat{\mathbf{s}}) = \underset{\mathbf{x}, \mathbf{s}}{\operatorname{argmax}} \log p(\mathbf{x}, \mathbf{s} | \mathbf{y}). \quad (7)$$

Interestingly, we showed in [1] that the solution of (7) corresponds (under mild conditions) to the solution of the standard (unstructured) SR problem (1) for a Bernoulli-Gaussian model *i.e.*, when model (3)-(5) is considered with $\mathbf{W} = 0_{M \times M}$. This result led us to the design of a new family of Bayesian pursuit algorithms. We developed in particular a Bayesian version of OMP, the Bayesian OMP (BOMP). Motivated by this connection between standard formulation and joint MAP estimation, we propose here to extend BOMP to a structured version using the generalization of Bernoulli model (6), namely the Boltzmann machine (5). In the sequel, we will thus refer to the proposed procedure as the “*Structured Bayesian Orthogonal Matching Pursuit*” (SBOMP) algorithm.

Note that our work distinguishes here from contributions [8, 10] in which the authors propose to solve a *marginalized* MAP estimation of the SR support \mathbf{s} , while considering the same Bayesian model (3)-(5).

4 Structured BOMP

SBOMP is a greedy algorithm looking for a solution of (7) via a succession of conditional maximizations. Formally, SBOMP generates a sequence of estimates $\{\hat{\mathbf{s}}^{(n)}, \hat{\mathbf{x}}^{(n)}\}_{n=1}^{\infty}$ defined as

$$\hat{s}_i^{(n)} = \begin{cases} \tilde{s}_i^{(n)} & \text{if } i = i^* \\ \hat{s}_i^{(n-1)} & \text{otherwise,} \end{cases} \quad (8)$$

$$\text{where } \tilde{s}_i^{(n)} = \underset{s_i}{\operatorname{argmax}} \{ \max_{x_i} \log p(\mathbf{x}, \mathbf{s} | \mathbf{y}) \} \quad (9)$$

$$\text{s. t. } (s_j, x_j) = (\hat{s}_j^{(n-1)}, \hat{x}_j^{(n-1)}) \quad \forall j \neq i,$$

$$\text{and } \hat{\mathbf{x}}^{(n)} = \underset{\mathbf{x}}{\operatorname{argmax}} \{ \log p(\mathbf{x}, \hat{\mathbf{s}}^{(n)} | \mathbf{y}) \}. \quad (10)$$

In a nutshell, SBOMP performs the following updates: at each iteration one single element of \mathbf{s} is updated, see (8); the update is based on a joint optimization of $\log p(\mathbf{x}, \mathbf{s} | \mathbf{y})$ with respect to (s_i, x_i) while other variables are kept fixed, see (9). Then, in a second step, \mathbf{x} is updated by taking the new support estimate $\hat{\mathbf{s}}^{(n)}$ into account, see (10).

We see in (8) that the index i^* of the element of \mathbf{s} which is updated must be specified. We choose to update the element which leads to the greatest increase

of the objective function, *i.e.*,

$$\begin{aligned}
 i^* &= \operatorname{argmax}_i \{ \max_{(s_i, x_i)} \log p(\mathbf{x}, \mathbf{s} | \mathbf{y}) \} \\
 \text{s. t. } &(s_j, x_j) = (\hat{s}_j^{(n-1)}, \hat{x}_j^{(n-1)}) \quad \forall j \neq i.
 \end{aligned} \tag{11}$$

Note that (9) and (10) corresponds to conditional maximizations of $\log p(\mathbf{x}, \mathbf{s} | \mathbf{y})$ (with respect to (x_i, s_i) and \mathbf{x} respectively). SBOMP thus defines a descent algorithm (the descent function being $-\log p(\mathbf{x}, \mathbf{s} | \mathbf{y})$). Moreover, $\hat{\mathbf{s}}^{(n)}$ (and therefore $\hat{\mathbf{x}}^{(n)}$) can only take on a finite number of values. Consequently, SBOMP is ensured to converge to a fixed point in a finite number of iterations.

In order to compare SBOMP to its standard unstructured version OMP, we give the expressions of update equations (8)-(11) particularized to probabilistic model (3)-(5) in Table 1. The update equations implemented by OMP are given in Table 2. Note that the formulation of OMP in Table 2 is slightly unconventional for the sake of comparison with SBOMP.

The solution of problem (9) is given in (12)-(13). $\tilde{s}_i^{(n)}$ corresponds to the value that will be assigned to \hat{s}_i if we decide to modify the i th component of $\hat{\mathbf{s}}$ at iteration n . We see in (12)-(13) that the value of $\tilde{s}_i^{(n)}$ is fixed via a threshold decision on a metric depending on the current residual error $\mathbf{r}^{(n-1)}$. The value of the threshold T_i depends on the decisions previously made on $\hat{s}_j^{(n-1)}$ $j \neq i$. The prior information on the structure of the sparse representation is therefore taken into account via a modification of the threshold T_i through the iterations. Note that the value of $\tilde{s}_i^{(n)}$ can be either 0 or 1. This implies that atom deselection is possible when $\hat{s}_i^{(n-1)} = 1$ and $\tilde{s}_i^{(n)} = 0$. The step corresponding to (12)-(13) in the OMP algorithm is the trivial operation (19), *i.e.*, OMP can only add atoms to the support of the sparse representation, irrespective of the decisions made during the previous iterations.

The choice of the element of $\hat{\mathbf{s}}$ modified at iteration n is given in (14)-(15). This corresponds to the solution of problem (11). The function optimized in (14) is made up of three terms which accounts for different effects. The first term weights the variation of the residual error if the i th component of the sparse representation is modified. It corresponds to the objective function considered by OMP in (20)-(21) when $\hat{s}_i^{(n-1)} = 0$, $\tilde{s}_i^{(n)} = 1$ and $\sigma_x^2 \rightarrow \infty$. The last two terms are not present in the OMP implementation, they stem from the structured probabilistic model considered in this paper: the second term accounts for the prior information available on x_i , it vanishes when $\sigma_x^2 \rightarrow \infty$, while the last term stands for the structure of the support of the sparse representation. It therefore depends on the previous decisions, $\hat{s}_i^{(n-1)}$'s, made on the support.

The support update equations (16) and (22) are identical for SBOMP and OMP. They model the fact that only one single element of the support estimate $\hat{\mathbf{s}}$ can be modified at each iteration. We remind however the reader that SBOMP update can lead to both atom selection and deselection whereas OMP is restricted to atom selections.

Finally, an explicit expression of SBOMP coefficient update (10) is given in (17)-(18). The corresponding operation is given in (23)-(24) for OMP. We can

<p>Initialization : $\mathbf{r}^{(0)} = \mathbf{y}$, $\hat{\mathbf{s}}^{(0)} = \mathbf{0}$ and $\hat{\mathbf{x}}^{(0)} = \mathbf{0}$.</p> <p>Repeat :</p> <p>1. $\forall i \in \{1, \dots, M\}$, evaluate</p> $\hat{s}_i^{(n)} = \begin{cases} 1 & \text{if } \langle \mathbf{r}^{(n-1)} + \hat{x}_i^{(n-1)} \mathbf{d}_i, \mathbf{d}_i \rangle^2 > T_i, \\ 0 & \text{otherwise,} \end{cases} \quad (12)$ <p>where $T_i = -2\sigma^2 \frac{\sigma^2 + \sigma_x^2}{\sigma_x^2} (b_i + 2 \sum_{j \neq i} \hat{s}_j^{(n-1)} w_{ij})$,</p> (13) <p>2. Choose the atom to be modified:</p> $i^* = \operatorname{argmin}_i \left\{ \frac{1}{2\sigma^2} \ \mathbf{r}^{(n-1)} + (\hat{x}_i^{(n-1)} - \hat{x}_i^{(n)}) \mathbf{d}_i\ _2^2 \right. \\ \left. - \frac{1}{2\sigma_x^2} (\hat{x}_i^{(n-1)2} - \hat{x}_i^{(n)2}) + (\hat{s}_i^{(n-1)} - \hat{s}_i^{(n)}) (b_i + 2 \sum_{j \neq i} \hat{s}_j^{(n-1)} w_{ij}) \right\}, \quad (14)$ <p>where $\hat{x}_i^{(n)} = \hat{s}_i^{(n)} \left(\hat{x}_i^{(n-1)} + \langle \mathbf{r}^{(n-1)}, \mathbf{d}_i \rangle \right) \frac{\sigma_x^2}{\sigma^2 + \sigma_x^2}$,</p> (15) <p>3. Update the SR support</p> $\forall i \in \{1, \dots, M\} \quad \hat{s}_i^{(n)} = \begin{cases} \hat{s}_i^{(n)} & \text{if } i = i^*, \\ \hat{s}_i^{(n-1)} & \text{otherwise.} \end{cases} \quad (16)$ <p>4. Update the SR coefficients</p> $\hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n)}}^{(n)} = \left(\mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T \mathbf{D}_{\hat{\mathbf{s}}^{(n)}} + \frac{\sigma^2}{\sigma_x^2} \mathbf{I}_{\ \hat{\mathbf{s}}^{(n)}\ _0} \right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T \mathbf{y}, \quad (17)$ <p>and $\forall i \in \{1, \dots, M\} \quad \hat{\mathbf{x}}_i^{(n)} = 0$ if $\hat{s}_i^{(n)} = 0$.</p> (18) <p>5. Update the residual: $\mathbf{r}^{(n)} = \mathbf{y} - \sum_{i=1}^M \hat{s}_i^{(n)} \hat{x}_i^{(n)} \mathbf{d}_i$.</p>

Table 1: Definition of the SBOMP algorithm

notice that SBOMP differs from OMP by the fact that it exploits the a priori variance σ_x^2 in the update of the coefficients. SBOMP and OMP coefficient updates reduce to the same operation when $\sigma_x^2 \rightarrow \infty$.

5 Experiments

In this section, we study the performance of SBOMP by extensive computer simulations. To that end, we study the ability of SBOMP to recover the coefficients of the sparse representation and measure the mean-square error (MSE) between the non-zero coefficients and their estimates.

The simulation data are generated according to observation model (3) and the prior model on \mathbf{x} (4). For an objective evaluation of the performance however, we build the SR support regardless the Boltzmann machine (5). Each point of simulation corresponds to a fixed number of non-zero coefficients and a particular combination of atoms. The indices of the atoms are thus drawn uniformly at random *once* for *all* observations. We use then the following pa-

<p>Initialization : $\mathbf{r}^{(0)} = \mathbf{y}$, $\hat{\mathbf{s}}^{(0)} = 0$.</p> <p>Repeat :</p> <ol style="list-style-type: none"> $\forall i \in \{1, \dots, M\}$, set $\tilde{s}_i^{(n)} = 1. \quad (19)$ Choose the atom to be modified: $i^* = \operatorname{argmax}_i \ \mathbf{r}^{(n-1)} - \tilde{x}_i^{(n)} \mathbf{d}_i\ _2^2, \quad (20)$ <p>where $\tilde{x}_i^{(n)} = \langle \mathbf{r}^{(n-1)}, \mathbf{d}_i \rangle$. (21)</p> Update the SR support $\forall i \in \{1, \dots, M\} \quad \hat{s}_i^{(n)} = \begin{cases} \tilde{s}_i^{(n)} & \text{if } i = i^*, \\ \hat{s}_i^{(n-1)} & \text{otherwise.} \end{cases} \quad (22)$ Update the SR coefficients $\hat{\mathbf{x}}_{\hat{\mathbf{s}}^{(n)}}^{(n)} = \left(\mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T \mathbf{D}_{\hat{\mathbf{s}}^{(n)}} \right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}^{(n)}}^T \mathbf{y}, \quad (23)$ <p>and $\forall i \in \{1, \dots, M\} \quad \hat{\mathbf{x}}_i^{(n)} = 0$ if $\hat{s}_i^{(n)} = 0$. (24)</p> <p>3. Update the residual: $\mathbf{r}^{(n)} = \mathbf{y} - \sum_{i=1}^M \hat{s}_i^{(n)} \hat{x}_i^{(n)} \mathbf{d}_i$.</p>
--

Table 2: Definition of the OMP algorithm

parameters: $M = 64$, $N = 32$, $\sigma^2 = 10^{-2}$, $\sigma_x^2 = 1$. The elements of the dictionary are generated for each observation as realizations of a zero-mean Gaussian distribution with variance N^{-1} . For each point of simulation, we consider 500 observations.

The parameters of the Boltzmann machine are drawn from the a posteriori distribution $p(\mathbf{b}, \mathbf{W}|\mathbf{s})$ by means of the ‘‘Single-variable Exchange’’ algorithm introduced in [12], using $w_{ij} \sim \mathcal{U}[-1, 1] \forall i, j$ and $b_i \sim \mathcal{U}[-20, 20] \forall i$. For each point of simulation, the ‘‘Single-variable Exchange’’ algorithm is run with a burn-out iteration number of 1000, we allocate then the 500 following parameter estimates for the 500 observations of the considered point.

SBOMP is compared to three other state-of-the-art algorithms: OMP, BOMP, which do not take into account any structure while looking for the sparse decompositions, and BM_MAP_OMP introduced by Faktor *et al.* in [10]. OMP is run until the ℓ_2 -norm of the residual drops below $\sqrt{N\sigma_n^2}$. The Bayesian algorithms BOMP and SBOMP iterate as long as $\log p(\mathbf{y}, \hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)}) > \log p(\mathbf{y}, \hat{\mathbf{x}}^{(n-1)}, \hat{\mathbf{s}}^{(n-1)})$.

Figures 1(a) and (b) show the MSE on the non-zero coefficients obtained for each of the four considered algorithms and two different setups: in figure (a), the variance σ_x^2 is supposed to be known while in figure (b), it is set to $\sigma_x^2 = 1000$ in the three Bayesian algorithms BOMP, SBOMP and BM_MAP_OMP to approach a non-informative prior $p(\mathbf{x})$. For both setups, SBOMP and BM_MAP_OMP outperform OMP and BOMP, confirming the relevance of accounting structures in sparse decompositions. All three Bayesian algorithms are susceptible to see its performance modified whether σ_x^2 is known or not, but we can see that the

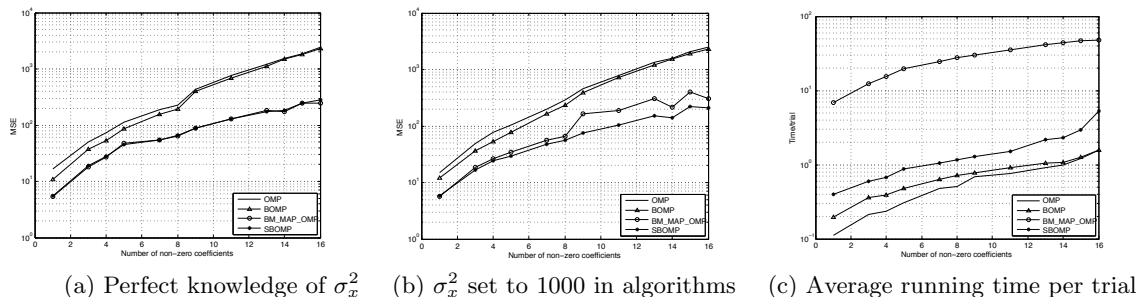


Figure 1: MSE on non-zero coefficients (figures (a) and (b)) and average running time (figure (c)) vs. the number of non-zero coefficients K .

performance of BM_MAP_OMP is more damaged than the one of BOMP and SBOMP.

Figure 1(c) presents the average running time per trial for each of the four considered algorithms. Not surprisingly, OMP and BOMP are the less costly procedures. More interesting is the large gap between the “structured” algorithms, SBOMP and BM_MAP_OMP. The latter relies on an iterative process which at each iteration requires the evaluation of a determinant which in high dimension can be very computationally demanding.

6 Conclusion

In this paper, we address the structured SR problem from a Bayesian point of view. Structures are taken into account by means of a Boltzmann machine which allows for the description of a large set of structures. We propose a greedy SR algorithm, SBOMP, which looks for the solution of a joint MAP problem by a sequence of conditional maximizations. SBOMP offers desirable features in comparison to OMP: it allows for atom deselection and takes the prior information about the structure of the sparse representation into account. We compare the performance of SBOMP to that of another state-of-the-art algorithm dealing with a Boltzmann machine [10]. If both “structured” algorithms have similar performance when the prior information on the sparse coefficient vector is known, the gap widens when considering a non-informative prior. Moreover, the proposed algorithm offers a better compromise between performance of reconstruction and computational cost, since its running time remains reasonable with regard to OMP and its unstructured homologue BOMP.

7 Acknowledgments

The authors wish to thank M. Tomer Faktor and Prof. Michael Elad for providing their implementation of the BM_MAP_OMP algorithm.

References

- [1] C. Herzet and A. Drémeau, “Bayesian pursuit algorithms,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, August 2010.
- [2] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proc. Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.
- [3] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67, 2006.
- [4] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, “Block-sparse signals: uncertainty relations and efficient recovery,” *IEEE Trans. On Signal Processing*, vol. 58, pp. 3042 – 3054, 2010.
- [5] M. Kowalski and B. Torr sani, “Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients,” *Signal, image and video processing*, vol. 3, no. 3, pp. 251–264, 2009.
- [6] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, “Proximal methods for hierarchical sparse coding,” Tech. Rep., INRIA, 2010.
- [7] C. F votte, L. Daudet, S. J. Godsill, and B. Torr sani, “Sparse regression with structured priors: application to audio denoising,” *IEEE Trans. On Audio, Speech and Language Processing*, vol. 16, no. 1, pp. 174–185, 2008.
- [8] P. J. Garrigues and B. A. Olshausen, “Learning horizontal connections in a sparse coding model of natural images,” in *Advances in Neural Information Processing Systems (NIPS)*, December 2008, pp. 505–512.
- [9] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, “Sparse signal recovery using markov random fields,” in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2008.
- [10] T. Faktor, Y. C. Eldar, and M. Elad, “Exploiting statistical dependencies in sparse representations for signal recovery,” Submitted to *IEEE Trans. On Signal Processing*.
- [11] C. Soussen, J. Idier, D. Brie, and J. Duan, “From bernoulli-gaussian deconvolution to sparse signal restoration,” Tech. Rep., CRAN/IRCCyN, January 2010.
- [12] I. Murray, Z. Ghahramani, and D. J. C. MacKay, “Mcmc for doubly-intractable distributions,” in *Proc. Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2006, pp. 359–366, AUAI Press.