Boltzmann machine and mean-field approximation for structured sparse decompositions

Angélique Drémeau, Cédric Herzet, Laurent Daudet

Boltzmann machine and mean-field approximation for structured sparse decompositions

Angélique Drémeau, Cédric Herzet and Laurent Daudet, *Senior Member, IEEE*

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 marginalized maximum a posteriori problem. To solve this problem, we resort to a mean-field approximation and the "variational Bayes Expectation-Maximization" algorithm. This approach results in a *soft* procedure making no hard decision on the support or the values of the sparse representation. We show that this characteristic leads to an improvement of the performance over state-of-the-art algorithms.

Index Terms

Structured sparse representation, Bernoulli-Gaussian model, Boltzmann machine, mean-field approximation.

I. 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. These decompositions have proved useful in a variety of domains including audio ([1], [2]) and image ([3], [4]) processing and are at the heart of the recent compressive-sensing paradigm [5].

A. Drémeau and L. Daudet are with Institut Langevin, ESPCI ParisTech, CNRS UMR 7587, France. A. Drémeau was supported by a fellowship from the Fondation Pierre-Gilles De Gennes pour la Recherche, France; she is currently working at Institut Télécom, Télécom ParisTech, CNRS LTCI, France. C. Herzet is with INRIA Centre Rennes - Bretagne Atlantique, France. L. Daudet is on a joint affiliation with Université Paris Diderot - Paris 7 and the Institut Universitaire de France.

Formally, let $\mathbf{y} \in \mathbb{R}^N$ be an observed signal and $\mathbf{D} \in \mathbb{R}^{N \times M}$ with $M \geq N$, a dictionary, *i.e.,* a matrix whose columns correspond to atoms. Then one standard formulation of the sparse representation problem can be written as

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}\mathbf{z}\|_2^2 \;\; \text{subject to } \|\mathbf{z}\|_0 \leq L, \tag{1}$$

or, in its Lagrangian version

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_0, \tag{2}$$

where $\|\mathbf{z}\|_0$ denotes the ℓ_0 pseudo-norm which counts the number of non-zero elements in $\mathbf{z}$ and L, $\lambda > 0$ are parameters specifying the trade-off between sparsity and distortion.

Finding the exact solution of (1)-(2) is an NP-hard problem [6], *i.e.,* it generally requires a combinatorial search over the entire solution space. Therefore, heuristic (but tractable) algorithms have been devised to deal with this problem. These algorithms are based on different strategies that we review in section I-A.

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 $\mathbf{z}$ can no longer be chosen independently from each other but must obey some (deterministic or probabilistic) inter-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 dealing with this problem (see section I-B).

In this paper, we propose a novel algorithm addressing the SR problem in the structured setup and consider the standard, non-structured setup as a particular case. The proposed algorithm is cast within a Bayesian inference framework and based on the use of a particular variational approximation as a surrogate to an optimal maximum a posteriori (MAP) decision. In order to properly place our work in the rich literature pertaining to SR algorithms, we briefly review hereafter some of the algorithms coping with the standard and the structured SR problems.

A. Standard Sparse representation algorithms

The origin of the algorithms addressing the standard sparse representation problem (1)-(2) traces back to the fifties, *e.g.*, in the field of statistical regression [7] and operational research [8], [9]. The algorithms available today in the literature can roughly be divided into four main families:

1) The algorithms based on problem relaxation: these procedures replace the ℓ_0-norm by an ℓ_p-norm (with $0 < p \leq 1$). This approximation leads to a relaxed problem which can be solved efficiently by

standard optimization procedures. Well-known instances of algorithms based on such an approach are the Basis Pursuit (BP) [10], Least Absolute Shrinkage and Selection Operator (LASSO) [11] or Focal Underdetermined System Solver (FocUSS) [12] algorithms.

2) The iterative thresholding algorithms: these procedures build up the sparse vector $\mathbf{z}$ by making a succession of thresholding operations. The first relevant work in this family was realized by Kingsbury et Reeves [13] who derive an iterative thresholding method with the aim at solving problem (2). However, their contribution is done without a clear connection to the objective function (2). We find a more explicit version of their results in [14] where Blumensath and Davies introduce the Iterative Hard Thresholding (IHT) algorithm. Daubechies *et al.* propose in [15] a similar procedure while replacing the ℓ_0-norm by the ℓ_1-norm. The resulting algorithm relies then on a soft thresholding operation.

3) The pursuit algorithms: these methods build up the sparse vector $\mathbf{z}$ by making a succession of greedy decisions. There exist many pursuit algorithms in the current literature. Among the most popular, we can cite Matching Pursuit (MP) [16], Orthogonal Matching Pursuit (OMP) [17] or Orthogonal Least Square (OLS) [18]. The latter algorithms do not allow for the selection of more than one atom per iteration. This limitation is avoided by more recent procedures like Stagewise OMP (StOMP) [19], Subspace Pursuit (SP) [20] or Compressive Sampling Matching Pursuit (CoSaMP) [21].

4) The Bayesian algorithms: these procedures express the SR problem as the solution of a Bayesian inference problem and apply statistical tools to solve it. They mainly distinguish by the prior model, the considered estimation problem and the type of statistical tools they apply to solve it. Regarding the choice of the prior, a popular approach consists in modeling $\mathbf{z}$ as a continuous random variable whose distribution has a sharp peak at zero and heavy tails (*e.g.*, Cauchy [22], Laplace [23], [24], t-Student [25], Jeffrey's [26] distributions). Another approach, recently gaining in popularity, is based on a prior made up of the combination of Bernoulli and Gaussian distributions ([27]–[36]). Different variants of Bernoulli-Gaussian (BG) models exist. A first approach, as considered in [27], [30], [31], [34], consists in assuming that the elements of $\mathbf{z}$ are independently drawn from Gaussian distributions whose variances are controlled by Bernoulli variables: a small variance enforces elements to be close to zero whereas a large one defines a non-informative prior on non-zero coefficients. Another model on $\mathbf{z}$ based on BG variables is as follows: the elements of the sparse vector are defined as the multiplication of Gaussian and Bernoulli variables. This model has been exploited in the contributions [28], [29], [32], [33] and will be considered in the present paper. These two distinct hierarchical BG models share a similar marginal

expression of the form:

$$p(\mathbf{z}) = \prod_{i=1}^{M} \left(p_i \mathcal{N}(0, \sigma_0^2) + (1 - p_i)\mathcal{N}(0, \sigma_1^2) \right), \tag{3}$$

where the p_i's are the parameters of the Bernoulli variables. While σ_0^2 can be tuned to any positive real value in the first BG model presented above, it is set to 0 in the second one. This marginal formulation is directly used in many contributions as in [35], [36].

B. Structured sparse representation algorithms

The algorithms dedicated to "standard" SR problems (1)-(2) do not assume any dependency between the non-zero elements of the sparse vector, *i.e.,* they select the atoms of the sparse decomposition without any consideration of possible links between them. Yet, recent contributions have shown the existence of structures in many natural signals (depending on the dictionary and the class of signals) and emphasize the relevance of exploiting them in the process of sparse decomposition. Hence, many contributions have recently focused on the design of "structured" sparse representation algorithms, namely algorithms taking the dependencies between the elements of SR support into account.

The algorithms available in the literature essentially rely on the same type of approximation as their standard counterpart (see section I-A) and could be classified accordingly. We found however more enlightening to present the state-of-the-art contributions according to the type of structure they exploit. We divide them into four families:

1) Group sparsity: in group-sparse signals, coefficients are either all non-zero or all zero within pre-specified groups of atoms. This type of structure is also referred to as *block sparsity* in some contributions ([37], [38]). In practice, group sparsity can be enforced by the use of particular "mixed" norms combining ℓ_1- and ℓ_2-norms. Following this approach, Yuan and Lin propose in [39] a LASSO-based algorithm called Group-LASSO, while in [37], Eldar and Mishali derive a modified SOCP (Second Order Cone Program) algorithm and in [38], Eldar *et al.* introduce the Block-OMP, group-structured extension of OMP. Parallel to these contributions, other approaches have been proposed. Let us mention [40] and [41] based on clusters, [42] where coding costs are considered, or [43] relying on the definition of Boolean variables and the use of an approximate message passing algorithm [44]. Finally, as an extension of group sparsity, Sprechmann *et al.* consider in [45] intra-group sparsity by means of an additional penalty term.

2) Molecular sparsity: molecular sparsity describes more complex structures, in the particular case where the atoms of the dictionary have a double indexing (*e.g.*, time-frequency atoms). It can be seen

as the combination of two group-sparsity constraints: one on each component of the double index. This type of structure is also referred to as *elitist sparsity* by certain authors [46].

In order to exploit molecular sparsity, Kowalski and Torrésani study in [46] the general use of mixed norms in structured sparsity problems. They thus motivate the Group-LASSO algorithm introduced in [39] and propose an extension of it, the Elitist-LASSO. Molecular sparsity has also been considered by Daudet in [47] for audio signals: the paper introduces the Molecular-MP algorithm which uses a local tonality index.

3) Chain and tree-structured sparsity: such structures arise in many applications. For example, chain structure appears in any sequential process whereas tree-structured sparsity is at the heart of wavelet decompositions, widely used in image processing. De facto, we find in the literature several contributions dealing with these particular types of constrained sparsity. Tree-structured sparsity is addressed in [48] where the authors define a particular penalty term replacing the commonly used ℓ_0- or ℓ_1-norms, and [49], [50] which define a probabilistic framework based on Bernoulli variables with scale-depending parameters. These two latter contributions focus on the sampling of the posterior distribution of $\mathbf{z}$ and resort either to Monte Carlo Markov Chain (MCMC) methods or to mean-field approximations. Chain-structured sparsity can be enforced using a Markov-chain process. This is for example the model adopted by Févotte *et al.* in [2], combined then with a MCMC inference scheme, and by Schniter in [51], together with an approximate message passing algorithm [44].

4) Generic structured sparsity: some 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, [52], [53] and [54] have recently emphasized the relevance of the Boltzmann machine as a general model for structured sparse representations. Well-known in Neural Networks, this model allows indeed to consider dependencies between distant atoms and thus constitutes an adaptive framework for the design of structured SR algorithms. In this paper, we will consider this particular model to derive a novel structured SR algorithm.

Finally, let us mention the deterministic approach in [55] which introduces the model-based CoSaMP, relying on the definition of a "model" peculiar to a structure. As practical examples, the authors apply their algorithm to group and tree-structured sparsity.

C. Contributions of this paper

In this paper, we focus on the design of an effective structured SR algorithm within a Bayesian framework. Motivated by a previous result [32], a Boltzmann machine is introduced to describe general

sparse structures. In this context, we reformulate the structured sparse representation problem as a particular marginalized maximum a posteriori (MAP) problem on the support of the sparse vector. We then apply a particular variational mean-field approximation to deal with the intractability of the original problem; this results in the so-called "SSoBaP" algorithm. We emphasize that SSoBaP shares some structural similarities with MP but enjoys additional desirable features: *i)* it can exploit a number of different structures on the support; *ii)* its iterative process is based on the exchange of *soft* decisions (by opposition to *hard* decisions for MP) on the support. We confirm through simulation results that SSoBaP leads to an improvement of the reconstruction performance (according to several figures of merits) over several SR algorithms of the state-of-the-art.

D. Organization of this paper

The paper is organized as follows. Section II describes the probabilistic model used to derive our algorithm. In particular, we suppose that the SR support is distributed according to a Boltzmann machine and show that this model allows to describe many well-known probabilistic model as particular cases.

In this framework, section III presents different Bayesian estimators which can be considered within the SR problematic. We focus in particular on a marginalized maximum a posteriori (MAP) problem on the SR support.

Section IV is dedicated to the resolution of this MAP problem. We propose in this paper to resort to a mean-field approximation and the "variational Bayes Expectation-Maximization" algorithm. The first subsection of section IV recalls the basics of this variational approach. The rest of the section is dedicated to the description of the proposed algorithm.

The performance of the proposed algorithm is evaluated in section V by various experiments involving different evaluation criteria on synthetic data. We show that, as long as our simulation setups are concerned, the proposed algorithm is very competitive with state-of-the-art procedures.

II. PROBABILISTIC MODEL

Let $\mathbf{x} \in \mathbb{R}^M$ be a vector defining the amplitudes of the sparse representation and $\mathbf{s} \in \{0,1\}^M$ be a vector defining the SR support, *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[1]:

$$\mathbf{y} = \sum_{i=1}^{M} s_i \, x_i \, \mathbf{d}_i + \mathbf{n}, \tag{4}$$

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

$$p(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \mathcal{N}(\mathbf{D_s x_s}, \sigma_n^2 \mathbf{I}_N), \tag{5}$$

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

$$p(\mathbf{x}) = \prod_{i=1}^{M} p(x_i) \;\; \text{where} \quad p(x_i) = \mathcal{N}(0, \sigma_{x_i}^2). \tag{6}$$

Within model (5)-(6), the observation $\mathbf{y}$ is thus seen as the noisy combination of atoms specified by $\mathbf{s}$. The weights of the combination are realizations of Gaussian distributions whose variances are independent on the support $\mathbf{s}$.

Clearly both the number of atoms building up $\mathbf{y}$ as well as their interdependencies are a function of the prior defined on $\mathbf{s}$. A standard choice for modelling *unstructured* sparsity is based on a product of Bernoulli distributions, *i.e.,*

$$p(\mathbf{s}) = \prod_{i=1}^{M} p(s_i) \;\; \text{where} \quad p(s_i) = \text{Ber}(p_i), \tag{7}$$

and $p_i \ll 1$. This model is indeed well-suited to modelling situations where $\mathbf{y}$ stems from a sparse process: if $p_i \ll 1 \;\forall\; i$, only a small number of s_i's will *typically*[2] 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 $\mathbf{D}$.

Note that (7) does not impose any interaction between the atoms building up the observed vector $\mathbf{y}$: each s_i is the realization of an independent random variable. Taking atom interdependencies into account therefore requires more involved probabilistic models. The so-called *Boltzmann machine* offers a nice option for this purpose [56]. Formally, it can be expressed as:

$$p(\mathbf{s}) \propto \exp(\mathbf{b}^T \mathbf{s} + \mathbf{s}^T \mathbf{W} \mathbf{s}), \tag{8}$$

[1]The sparse representation $\mathbf{z}$, as used in section I, is then defined as the Hadamard product of $\mathbf{x}$ and $\mathbf{s}$, *i.e.,* $z_i = s_i x_i, \;\forall i \in [\![1, M]\!]$.

[2]In an information-theoretic sense, *i.e.,* according to model (5)-(7), a realization of $\mathbf{s}$ with a few non-zero components will be observed with probability almost 1.

where $\mathbf{W}$ is a symmetric matrix with zeros on the diagonal and $\propto$ denotes equality up to a normalization factor. Parameter $\mathbf{b}$ defines the biases peculiar to each element of $\mathbf{s}$ while $\mathbf{W} \in \mathbb{R}^{M \times M}$ characterizes the interactions between them: w_{ij} weights the dependency between atoms s_i and s_j.

The Boltzmann machine encompasses many well-known probabilistic models as particular cases. For example, the Bernoulli model (7) corresponds to $\mathbf{W} = \mathbf{0}_{M \times M}$ (expressing the atoms' independence):

$$p(\mathbf{s}) \propto \exp(\mathbf{b}^T \mathbf{s}) = \prod_i \exp(b_i s_i), \tag{9}$$

which is equivalent to a Bernoulli model (7) with

$$p_i = \frac{1}{1 + \exp(-b_i)}. \tag{10}$$

Another example is the Markov chain. For instance, let us consider the following first-order Markov chain:

$$p(\mathbf{s}) = p(s_1) \prod_{i=1}^{M} p(s_{i+1}|s_i), \tag{11}$$

with $\forall i \in [\![1, M-1]\!]$,

$$p(s_{i+1} = s | s_i = 1) \triangleq \begin{cases} 1 - p_{i+1}^1 & \text{if } s = 1, \\ p_{i+1}^1 & \text{if } s = 0, \end{cases} \tag{12}$$

$$p(s_{i+1} = s | s_i = 0) \triangleq \begin{cases} p_{i+1}^0 & \text{if } s = 1, \\ 1 - p_{i+1}^0 & \text{if } s = 0, \end{cases} \tag{13}$$

$$p(s_1 = s) \triangleq \begin{cases} p_1 & \text{if } s = 1, \\ 1 - p_1 & \text{if } s = 0. \end{cases} \tag{14}$$

This Markov chain corresponds to a Boltzmann machine with parameters $\mathbf{b}$ and $\mathbf{W}$ defined in (15)-(16) below. In particular, only two subdiagonals in $\mathbf{W}$ are non-zero.

$$\mathbf{b} = [\log \frac{p_1}{1-p_1} + \log \frac{p_2^1}{1-p_2^0} \ \ldots \ \log \frac{p_i^0}{1-p_i^0} + \log \frac{p_{i+1}^1}{1-p_{i+1}^0} \ \ldots \ \log \frac{p_M^0}{1-p_M^0}]^T, \tag{15}$$

$$\mathbf{W} = \begin{pmatrix} 0 & \frac{1}{2}\log\frac{(1-p_2^1)(1-p_2^0)}{p_2^1\,p_2^0} & 0 & 0 \\ \frac{1}{2}\log\frac{(1-p_2^1)(1-p_2^0)}{p_2^1\,p_2^0} & 0 & \frac{1}{2}\log\frac{(1-p_3^1)(1-p_3^0)}{p_3^1\,p_3^0} & \vdots \\ 0 & \frac{1}{2}\log\frac{(1-p_3^1)(1-p_3^0)}{p_3^1\,p_3^0} & \ddots & \vdots \\ \vdots & \ddots & 0 & \vdots \\ 0 & \ldots & \ldots & 0 \end{pmatrix}. \tag{16}$$

In the rest of this paper, we will derive the main equations of our algorithm for the general model (8). We will then particularize them to model (7), which leads to an algorithm for standard (unstructured) sparse representation.

III. Sparse representations within a Bayesian framework

The probabilistic framework defined in section II allows us to tackle the SR problem from a Bayesian perspective. As long as (5)-(6) 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 maximum a posteriori (MAP) criterion, which corresponds to the optimal Bayesian estimator for a Bayesian cost based on a "notch" loss function [57].

A first possible approach consists in solving the *joint* MAP problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{s}}) = \underset{\mathbf{x},\mathbf{s}}{\operatorname{argmax}} \log p(\mathbf{x}, \mathbf{s}|\mathbf{y}). \tag{17}$$

Interestingly, we emphasize in [32] that the joint MAP problem (17) shares the same set of solutions as the standard SR problem (2) within BG model (6)-(7). This connection builds a bridge between standard and Bayesian SR procedures and motivates the use of model (6)-(7) (and its structured generalization (6)-(8)) in other estimation problems. In particular, we focus hereafter on MAP problems oriented to the recovery of the SR support.

Assuming (5)-(6) is the true generative model of $\mathbf{y}$, the decision minimizing the probability of wrong decision on the *whole* SR support is given by

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\{0,1\}^M}{\operatorname{argmax}} \log p(\mathbf{s}|\mathbf{y}), \tag{18}$$

where $p(\mathbf{s}|\mathbf{y}) = \int_{\mathbf{x}} p(\mathbf{x}, \mathbf{s}|\mathbf{y}) d\mathbf{x}$. Problem (18) is unfortunately intractable since it typically requires the

evaluation of the cost function, $\log p(\mathbf{s}|\mathbf{y})$, for all possible 2^M sequences in $\{0,1\}^M$. A heuristic greedy procedure looking for the solution of (18) has recently been proposed in [54].

In this paper, we address the SR representation problem from a different perspective. The decision on each element of the support is made from a marginalized MAP estimation problem:

$$\hat{s}_i = \underset{s_i \in \{0,1\}}{\operatorname{argmax}} \log p(s_i|\mathbf{y}). \tag{19}$$

The solution of (19) minimizes the probability of making a wrong decision on *each* s_i (rather than on the whole sequence as in (18)).

At first sight, problem (19) may appear easy to solve since the search space only contains two elements *i.e.,* $s_i \in \{0,1\}$. However, the evaluation of $p(s_i|\mathbf{y})$ turns out to be intractable since it requires a costly marginalization of the joint probability $p(\mathbf{s}|\mathbf{y})$ over the s_j's, $j \neq i$. Nevertheless, many tools exist in the literature to circumvent this issue. In particular, the family of variational approximations allows for the computation of tractable surrogates of $p(s_i|\mathbf{y})$, say $q(s_i)$, see [58]. In this paper we will resort to a *mean-field* variational approximation to compute a surrogate $q(s_i)$ of $p(s_i|\mathbf{y})$ (see next section). In particular, in this paper we will resort to a mean-field variational approximation to compute a tractable surrogate of $p(s_i|\mathbf{y})$, say $q(s_i)$. Problem (19) will then be approximated by

$$\hat{s}_i = \underset{s_i \in \{0,1\}}{\operatorname{argmax}} \log q(s_i), \tag{20}$$

which is straightforward to solve.

Finally, given the estimated support $\hat{\mathbf{s}}$, we can reconstruct the coefficients of a sparse representation say $\hat{\mathbf{x}}_{\hat{\mathbf{s}}}$, as its MAP estimate

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \log p(\mathbf{x}|\hat{\mathbf{s}}, \mathbf{y}). \tag{21}$$

The solution of (21) is expressed as

$$\hat{\mathbf{x}}_{\hat{\mathbf{s}}} = \left(\mathbf{D}_{\hat{\mathbf{s}}}^T \mathbf{D}_{\hat{\mathbf{s}}} + \Delta\right)^{-1} \mathbf{D}_{\hat{\mathbf{s}}}^T \mathbf{y}, \tag{22}$$

$$\text{and} \quad \hat{x}_i = 0 \ \text{ if } \ s_i = 0,$$

where Δ is a diagonal matrix whose ith element is $\frac{\sigma_n^2}{\sigma_{x_i}^2}$. When $\sigma_{x_i}^2 \to +\infty \ \forall i$, (22) reduces to the

least-square estimate

$$\hat{\mathbf{x}}_{\hat{\mathbf{s}}} = \mathbf{D}_{\hat{\mathbf{s}}}^{+}\mathbf{y}, \tag{23}$$

$$\text{and} \quad \hat{x}_i = 0 \ \text{ if } \ s_i = 0,$$

where $\mathbf{D}_{\hat{\mathbf{s}}}^{+}$ is the Moore-Penrose pseudo-inverse of the matrix made up of the $\mathbf{d}_i$'s such that $\hat{s}_i = 1$.

IV. Structured Soft Bayesian Pursuit Algorithm

In this section, we detail our methodology to compute the approximation $q(s_i)$ of the posterior probability $p(s_i|\mathbf{y})$. Our approach is based on a well-known variational approximation, namely the mean-field (MF) approximation, and its practical implementation via the so-called VB-EM algorithm. This methodology results in an iterative algorithm whose updates are very similar to those of the recently-proposed Bayesian Matching Pursuit algorithm (BMP) [32]. However, unlike the latter, the proposed procedure updates probabilities rather than estimates of the SR support. Moreover, BMP as introduced in [32] does not deal with structured sparsity. In the sequel, we will thus refer to the proposed procedure as the "Structured Soft Bayesian Pursuit algorithm" (SSoBaP).

The rest of this section is organized as follows. We first briefly recall the general theory pertaining to mean-field approximations. Then, in subsection IV-B we derive the main equations defining SSoBaP. The subsection IV-C is dedicated to the Soft Bayesian Pursuit algorithm (SoBaP), particular case of SSoBaP resulting from the choice $\mathbf{W} = \mathbf{0}_{M\times M}$ in the Boltzmann machine (8). In the next subsection, we emphasize the advantage of making soft decisions by comparing the update equations of BMP and SSoBaP. We address the problem of parameter estimation in a "variational Bayes Expectation-Maximization" framework in subsection IV-E and finally, emphasize the differences and connections of SSoBaP (and SoBaP) with existing algorithms in the last subsection.

A. Mean-field approximation: basics

The mean-field approximation [59] refers to a family of approximations of posterior probabilities by distributions having a "tractable" factorization. Formally, let $\boldsymbol{\theta}$ denote a vector of random variables and $p(\boldsymbol{\theta}|\mathbf{y})$ its a posteriori probability. Let moreover $(\boldsymbol{\theta}_i)_{i=1}^{I}$ denotes a *partition* of the elements of $\boldsymbol{\theta}$ *i.e.,*

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T \dots \boldsymbol{\theta}_I^T]^T. \tag{24}$$

Then, the mean-field approximation of $p(\boldsymbol{\theta}|\mathbf{y})$ relative to partition (24) is the surrogate distribution $q^\star(\boldsymbol{\theta})$ satisfying

$$q^\star(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta})}{\operatorname{argmin}} \left\{ \int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \log \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} \right) d\boldsymbol{\theta} \right\}, \tag{25}$$

subject to

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{I} q(\boldsymbol{\theta}_i), \qquad \int_{\boldsymbol{\theta}_i} q(\boldsymbol{\theta}_i)\, d\boldsymbol{\theta}_i = 1 \quad \forall i \in [\![1, I]\!]. \tag{26}$$

The mean-field approximation $q^\star(\boldsymbol{\theta})$ is therefore the distribution minimizing the Kullback-Leibler divergence with the actual posterior $p(\boldsymbol{\theta}|\mathbf{y})$ while factorizing as a product of probabilities (26). There potentially[3] are as many possible mean-field approximations as partitions of $\boldsymbol{\theta}$. In practice, the choice of a particular approximation results from a trade-off between complexity and accuracy.

A solution to problem (25)-(26) can be looked for by successively minimizing the Kullback-Leibler divergence with respect to one single factor, say $q(\boldsymbol{\theta}_i)$. This gives rise to the following update equations

$$\begin{aligned}
q^{(n+1)}(\boldsymbol{\theta}_1) &\propto \exp\left\{ \langle \log p(\boldsymbol{\theta}, \mathbf{y}) \rangle_{\prod_{j\neq 1} q^{(n)}(\boldsymbol{\theta}_j)} \right\}, \\
&\vdots \\
q^{(n+1)}(\boldsymbol{\theta}_i) &\propto \exp\left\{ \langle \log p(\boldsymbol{\theta}, \mathbf{y}) \rangle_{\substack{\prod_{j>i} q^{(n)}(\boldsymbol{\theta}_j) \\ \prod_{j<i} q^{(n+1)}(\boldsymbol{\theta}_j)}} \right\}, \\
&\vdots \\
q^{(n+1)}(\boldsymbol{\theta}_I) &\propto \exp\left\{ \langle \log p(\boldsymbol{\theta}, \mathbf{y}) \rangle_{\prod_{j\neq I} q^{(n+1)}(\boldsymbol{\theta}_j)} \right\},
\end{aligned} \tag{27}$$

where

$$\langle \log p(\boldsymbol{\theta}, \mathbf{y}) \rangle_{q(\boldsymbol{\theta}_i)} \triangleq \int_{\boldsymbol{\theta}_i} q(\boldsymbol{\theta}_i) \log p(\boldsymbol{\theta}, \mathbf{y}) d\boldsymbol{\theta}_i. \tag{28}$$

Note that we suppose in (27) that the $q(\boldsymbol{\theta}_i)$'s are updated at each iteration one after the other, in an increasing order of their indices. However the extension to other update schedulings is straightforward.

The procedure described in (27) is usually refered to as *"variational Bayes Expectation-Maximization (VB-EM) algorithm"* in the literature [60]–[62]. VB-EM is ensured to converge to a saddle point or a (local or global) maximum of problem (25)-(26) under mild conditions.

[3]Two different partitions can indeed lead incidentally to the same solution for (25)-(26).

The appellation "VB-EM" comes from the close connection of the above procedure with the well-known Expectation-Maximization (EM) algorithm [63]. The relation between the two algorithms can be seen by imposing an additional constraint on some $q(\boldsymbol{\theta}_i)$'s, namely

$$q(\boldsymbol{\theta}_i) = \delta(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i), \tag{29}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Minimizing the Kullback-Leibler divergence with respect to $q(\boldsymbol{\theta}_i)$ while taking (29) into account then reduces to optimizing the value of $\hat{\boldsymbol{\theta}}_i$. Thus, for the $\boldsymbol{\theta}_i$'s subject to (29), the update equation (27) can be rewritten as

$$\hat{\boldsymbol{\theta}}_i^{(n+1)} = \underset{\boldsymbol{\theta}_i}{\operatorname{argmax}} \left\{ \langle \log p(\boldsymbol{\theta}, \mathbf{y}) \rangle_{\substack{\prod_{j>i} q^{(n)}(\boldsymbol{\theta}_j) \\ \prod_{j<i} q^{(n+1)}(\boldsymbol{\theta}_j)}} \right\}. \tag{30}$$

Now, let $\boldsymbol{\theta}_{\neq i}$ denote the vector made of the $\boldsymbol{\theta}_j$'s, $j \neq i$. If only one element in the partition $(\boldsymbol{\theta}_i)_{i=1}^I$, say $\boldsymbol{\theta}_j$, is *not* subject to (29), it can be shown [64] that the update equations (27)-(30) define an EM algorithm aiming at solving

$$\hat{\boldsymbol{\theta}}_{\neq j} = \underset{\boldsymbol{\theta}_{\neq j}}{\operatorname{argmax}} \log p(\boldsymbol{\theta}_{\neq j}|\mathbf{y}), \tag{31}$$

where $\boldsymbol{\theta}_j$ is considered as a hidden variable. The E-step then corresponds to the estimation of $q(\boldsymbol{\theta}_j)$ (27), namely $p(\boldsymbol{\theta}_j|\mathbf{y}, \boldsymbol{\theta}_{\neq j})$ in this particular case, while the M-step computes (30) $\forall i \neq j$, *i.e.,* maximizes expectation $E_{\boldsymbol{\theta}_j}[\log p(\boldsymbol{\theta}, \mathbf{y})]$ with respect to parameters $\hat{\boldsymbol{\theta}}_{\neq j}$.

The general case where several $\boldsymbol{\theta}_i$'s are not subject to (29) (as opposed to the case presented above where all $\boldsymbol{\theta}_i$ but one where subject to (29)) does not correspond to an EM algorithm anymore as the E-step does not reduce to the estimation of *one* posterior probability but *approximates* a joint probability by means of an MF approximation.

To conclude this section, let us point out that mean-field approximations offer a nice framework to approximate the marginals $p(\boldsymbol{\theta}_i|\mathbf{y})$'s, where $\boldsymbol{\theta}_i$ is an element of the mean field partition (24) (note that we use here the word "marginal" in a large sense since $\boldsymbol{\theta}_i$ possibly contains more than one variable). Indeed, assume one wants to compute

$$p(\boldsymbol{\theta}_i|\mathbf{y}) = \int_{\boldsymbol{\theta}_{\neq i}} p(\boldsymbol{\theta}|\mathbf{y})\, d\boldsymbol{\theta}_{\neq i}. \tag{32}$$

Then, using the decomposition property of the mean-field approximation (26), we come up with

$$p(\boldsymbol{\theta}_i|\mathbf{y}) \simeq \int_{\boldsymbol{\theta}_{\neq i}} q(\boldsymbol{\theta})\, d\boldsymbol{\theta}_{\neq i} \tag{33}$$

$$\simeq q(\boldsymbol{\theta}_i) \int_{\boldsymbol{\theta}_{\neq i}} \prod_{j\neq i} q(\boldsymbol{\theta}_j) d\boldsymbol{\theta}_{\neq i} = q(\boldsymbol{\theta}_i). \tag{34}$$

Factors $q(\boldsymbol{\theta}_i)$'s are therefore approximations of marginals. We will exploit this observation in the next section to derive a tractable approximation of $p(s_i|\mathbf{y})$.

B. SSoBaP

In this paper, we consider the particular case where the MF approximation of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$, say $q(\mathbf{x}, \mathbf{s})$, is constrained to have the following structure:

$$q(\mathbf{x}, \mathbf{s}) = \prod_i q(x_i, s_i). \tag{35}$$

This is equivalent to setting $I = M$, $\boldsymbol{\theta} = [x_1 s_1 \dots x_M s_M]^T$ and $\boldsymbol{\theta}_i = [x_i\ s_i]^T\ \forall i$ in the general framework described in section IV-A. Note that, the $\boldsymbol{\theta}_i$'s do not correspond to single elements of $\boldsymbol{\theta}$ but form a partition of $\boldsymbol{\theta}$.

Particularized to model (5)-(6)-(8), the corresponding VB-EM update equations (27) are written as[4]:

$$q(x_i, s_i) = q(x_i|s_i)\ q(s_i), \tag{36}$$

where

$$q(x_i|s_i) = \mathcal{N}(m(s_i), \Sigma(s_i)), \tag{37}$$

$$q(s_i) \propto \sqrt{\Sigma(s_i)} \exp\left(\frac{1}{2}\frac{m(s_i)^2}{\Sigma(s_i)}\right) \tag{38}$$

$$\exp\left(s_i(b_i + 2\sum_{j\neq i} q(s_j = 1)\, w_{ij})\right),$$

[4]When clear from the context, we will drop the iteration indices for notational simplicity.

and

$$\Sigma(s_i) = \frac{\sigma_{x_i}^2 \sigma_n^2}{\sigma_n^2 + s_i \sigma_{x_i}^2 \mathbf{d}_i^T \mathbf{d}_i}, \tag{39}$$

$$m(s_i) = s_i \frac{\sigma_{x_i}^2}{\sigma_n^2 + s_i \sigma_{x_i}^2 \mathbf{d}_i^T \mathbf{d}_i} \langle \mathbf{r}_i \rangle^T \mathbf{d}_i, \tag{40}$$

$$\langle \mathbf{r}_i \rangle = \mathbf{y} - \sum_{j \neq i} q(s_j = 1)\, m(s_j = 1)\, \mathbf{d}_j. \tag{41}$$

After convergence of the procedure defined in (36)-(41), probabilities $q(x_i, s_i)$ correspond to a mean-field approximation of $p(x_i, s_i|\mathbf{y})$ (see (34)). Coming back to problem (19), an approximation of $p(s_i|\mathbf{y})$ thus simply follows from the relations:

$$p(s_i|\mathbf{y}) = \int p(x_i, s_i|\mathbf{y})\, dx_i, \tag{42}$$

$$\simeq \int q(x_i, s_i)\, dx_i = q(s_i). \tag{43}$$

This approximation can be used in problem (20) to make an approximated MAP decision on s_i. Note that (20) is easy to solve by simple thresholding operation, *i.e.,* $\hat{s}_i = 1$ if $q(s_i = 1) > T$ and $\hat{s}_i = 0$ otherwise, with $T = 0.5$.

The most expensive operation is the update equation (41) which scales as $\mathcal{O}(NM)$. So, the complexity of one update step is equal to Matching Pursuit (MP). However, in MP *one* unique couple (x_i, s_i) is involved at each iteration while in the proposed algorithm *all* indices are updated one after the other. To the extent of our experiments (see section V), we could observe that the proposed algorithm converges in a reasonable number of iterations, keeping it at a competitive position beside state-of-the-art algorithms.

C. A particular case: SoBaP

As emphasized in section II, the Boltzmann machine can be seen as a general framework including a large set of probabilistic models. Among them, the Bernoulli model (7) is of particular interest, as a possible approach to model unstructured sparsity (see *e.g.*, [32], [33]).

From a mathematical point of view, the Bernoulli model (7) corresponds to the simple case $\mathbf{W} = \mathbf{0}_{M \times M}$ in the Boltzmann machine (8). In this case, procedure (36)-(41) remains unchanged, except for (38) which becomes:

$$q(s_i) \propto \sqrt{\Sigma(s_i)} \exp\left(\frac{1}{2} \frac{m(s_i)^2}{\Sigma(s_i)}\right) p(s_i), \tag{44}$$

with $p(s_i) = \mathrm{Ber}(p_i)$, $\forall i \in [\![1, M]\!]$.

As the BG model (6)-(7) is largely used to address the unstructured SR problem, it is useful to distinguish the procedure using (44) from the SSoBaP process. To this end, we will refer to this particular case as "Soft Bayesian Pursuit algorithm" (SoBaP) in the sequel. Note that SoBaP was introduced from a BG perspective in our conference paper [65].

D. Soft versus hard decision

Contrarily to many deterministic (*e.g.*, [55] for structured sparsity, [10], [16], [17], [19] for unstructured sparsity) and probabilistic (*e.g.*, [52]–[54] for structured sparsity, [32], [33], [66] for unstructured sparsity) algorithms in the literature, the procedure defined in (36)-(41) does not make any hard decision on the SR support or the values of the SR coefficients at each iteration, but evaluates probabilities. It thus allows, to some extent, to take into account the uncertainties we have on the model and to refine this model at each iteration before making the final decision. In particular, it is worth comparing the proposed procedure to the Bayesian Matching Pursuit (BMP) introduced in [32] for unstructured sparsity.

BMP is an iterative procedure looking sequentially for a solution of (17). It proceeds like its standard homologue MP by modifying one unique couple (x_i, s_i) at each iteration, namely the one leading to the highest increase of $\log p(\mathbf{x}, \mathbf{s}|\mathbf{y})$. It can then be shown (see [32]) that the (locally) optimal update of the selected coefficient x_i is given by

$$\hat{x}_i^{(n)} = \hat{s}_i^{(n)} \frac{\sigma_{x_i}^2}{\sigma_n^2 + \sigma_{x_i}^2 \mathbf{d}_i^T \mathbf{d}_i} \mathbf{r}_i^{(n)T} \mathbf{d}_i, \tag{45}$$

$$\text{where} \qquad \mathbf{r}_i^{(n)} = \mathbf{y} - \sum_{j \neq i} \hat{s}_j^{(n-1)} \hat{x}_j^{(n-1)} \mathbf{d}_j, \tag{46}$$

and n is the iteration number. We omit here deliberately the support update, addressed in BMP from an "unstructured" point of view.

BMP and SSoBaP share some similarities. In particular, the mean of distribution $q(x_i|s_i)$ computed by the proposed algorithm (40) has the same form as the coefficient update performed by BMP (45). They rely however on different variables, namely the residual $\mathbf{r}_i$, (46), and its mean $\langle \mathbf{r}_i \rangle$, (41). This fundamental difference between both algorithms leads to well distinct approaches. In BMP, a hard decision is made on the SR support at each iteration: the atoms of the dictionary are either used or not (each $\hat{x}_j^{(n-1)}$ is multiplied by $\hat{s}_j^{(n-1)}$ which is equal to 0 or 1). On the contrary, in the proposed algorithm, the contributions of the atoms are simply weighted by $q(s_j = 1)$, *i.e.,* the probability distributions of the s_j's. In a similar way, the coefficients $\hat{x}_j^{(n-1)}$'s used in (46) are replaced by their means $m(s_j = 1)$ in (41), taking into account the uncertainties we have on the values of the x_j's.

E. Estimation of the noise variance

The estimation of model parameters can be naturally implemented in SSoBaP by procedure (29)-(30) described in section IV-A. Considering a set of unknown parameters $\boldsymbol{\alpha}$, one can include $q(\boldsymbol{\alpha})$ as a new factor within the VB-EM equations and possibly add the additional constraint

$$q(\boldsymbol{\alpha}) = \delta(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\,). \tag{47}$$

In the sequel, we will however not consider the *general* problem of model-parameter estimation, which can be particularly involved in Boltzmann machine. A lot of literature has already been dedicated to this problem and is out of the scope of this paper. We refer the interested reader to *e.g.*, [52], [54], [67]–[69].

In this section, we exclusively focus on the estimation of the noise variance σ_n^2 which has revealed to be crucial for the algorithm performance in our empirical experiences. The noise variance can be seen as a disparity measure between the observation $\mathbf{y}$ and its sparse approximation. Even if it is known *a priori*, its estimation turns out to be of great interest for the algorithm convergence. Indeed, SSoBaP relies on a successive refinement of the approximations of the posterior distributions $p(x_i, s_i|\mathbf{y})$'s, *i.e.,* the sparse approximation: in the first iterations, the estimations are likely to be coarse, thus the disparity between $\mathbf{y}$ and its sparse approximation might be large. The estimation of σ_n^2 at each iteration allows to take this evolution into account in the approximation process.

In practice, particularized to model (5)-(6)-(8), we consider σ_n^2 as a new unknown variable in $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = [x_1\ s_1\ \ldots\ x_M\ s_M\ \sigma_n^2]^T, \tag{48}$$

and add a factor in the MF structure (49) as

$$q(\mathbf{x}, \mathbf{s}, \sigma_n^2) = q(\sigma_n^2) \prod_i q(x_i, s_i). \tag{49}$$

Then, $q(\sigma_n^2)$ is constrained to

$$q(\sigma_n^2) = \delta(\sigma_n^2 - \hat{\sigma}_n^2), \tag{50}$$

leading to maximization (30), which becomes

$$\hat{\sigma}_n^2 = \underset{\sigma_n^2}{\operatorname{argmax}} \left\{ \sum_{\mathbf{s}} \int \prod_i q(x_i, s_i) \log p(\mathbf{x}, \mathbf{s}, \mathbf{y}|\sigma_n^2)\, d\mathbf{x} \right\}, \tag{51}$$

$$= \frac{1}{N} \left\langle \|\mathbf{y} - \sum_i s_i x_i \mathbf{d}_i\|^2 \right\rangle_{\prod_i q(x_i,s_i)} \tag{52}$$

$$\begin{aligned} &= \frac{1}{N} \Big(\mathbf{y}^T\mathbf{y} - 2\sum_i q(s_i=1) m(s_i=1) \mathbf{y}^T\mathbf{d}_i \\ &+ \sum_i \sum_{j\neq i} q(s_i=1) q(s_j=1) m(s_i=1) m(s_j=1) \mathbf{d}_i^T\mathbf{d}_j \\ &+ \sum_i q(s_i=1)(\Sigma(s_i=1) + m(s_i=1)^2) \mathbf{d}_i^T\mathbf{d}_i \Big). \end{aligned} \tag{53}$$

Update equation (53) is inserted in procedure (36)-(41) after the estimation of the $q(x_i, s_i)$'s.

F. Relation to past work

In this subsection, we place the SSoBaP algorithm and its particular "unstructured" case, SoBaP, within the previous contributions of the literature. To be as exhaustive as possible, we identify the contributions considering Boltzmann machines, but also those using BG models, in a SR context:

1) Boltzmann machine: The proposed SSoBaP can be compared to the three main contributions [52], [53] and [54], which consider Boltzmann machines in a structured SR point of view. They mainly distinguish by the estimation problem they consider and the practical procedure they propose to solve it.

In [52], the authors focus on the MAP estimation of the support of the sparse representation (18) and propose a solution using Gibbs sampling and simulated annealing. The same estimation problem is considered by Faktor *et al.* in [54]. Emphasizing the high computational cost of the approach [52], they suggest a greedy alternative. The greedy approach is also adopted in [53] but to solve the joint MAP estimation problem (17). In this contribution, the authors derive the so-called LaMP (for "Lattice Matching Pursuit"), a structured version of CoSaMP.

In next section V, we compare the proposed algorithm to the contributions [54], which presents a reasonable computational cost.

2) BG model: As mentioned in the introduction, BG model (6)-(7) has already been considered in some contributions ([28], [29], [32], [33]) and under the marginal formulation (3) in [35], [36]. However all these contributions differ from the proposed approach by the estimation problem and the practical procedure introduced to solve it.

Thus, in [28], [29], [32], [33], the authors focus on the joint MAP estimation problem (17). They then propose different greedy procedures to solve it, some of them are explicitly related to standard deterministic algorithms, as BMP, BOMP [32] or SBR [33] and their respective standard homologues MP, OMP and OLS.

Contribution [50] considers a tree-structured version of BG model (6)-(7) dedicated to a specific application (namely, the sparse decomposition of an image in wavelet or DCT bases). Besides this specific application, their approach relies, as ours, on a VB-EM algorithm. However, it differs by the MAP estimation problem (18) they address and the different MF factorization they choose to solve it. Finally, Ge *et al.* suggest in [34] another approximation of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$ based on a MCMC inference scheme.

In contributions [35], [36], the authors use the marginal formulation (3). They propose to resort to the approximate message passing algorithm introduced in [44] and generalized in [70] to compute the posterior distribution of the sought sparse vector. Both also consider the possibility of estimating the parameters of the model (*e.g.*, the noise variance, the Bernoulli parameter, the variance of the Gaussian distribution, etc.) by means of an Expectation-Maximization-like algorithm.

V. Experiments

In this section, we study the performance of the proposed algorithm by extensive computer simulations. We assess the performance in terms of the reconstruction of the SR support and the estimation of the non-zero coefficients. To that end, we evaluate different figures of merit as a function of the number of atoms used to generate the data, say K. In particular, we consider empirical measures of the mean square error (MSE), the probability of missed detections, the probability of false detections. These figures are evaluated from 500 trials for each simulation points.

We assess the performance of the proposed algorithm in both the unstructured and structured cases and compare the results to those obtained with state-of-the-art procedures.

A. Unstructured case

The unstructured case does not consider the possible structures existing between the atoms building the sparse representation. We use the following parameters: $N = 128$, $M = 256$, $\sigma_n^2 = 10^{-3}$ and generate the data as follows. Each point of simulation corresponds to a fixed number of non-zero coefficients K and, given this number, the positions of the non-zero coefficients are drawn uniformly at random for *each* observation. The elements of the dictionary are generated for each observation as realizations of

a zero-mean Gaussian distribution with variance N^{-1}. The value of the non-zero coefficients in $\mathbf{x}$ are generated according to the two different scenarios that we describe below.

We evaluate and compare the performance of 7 different algorithms: MP [16], SP [20], IHT [14], BP [10], BMP [32], EMBGAMP [36] and SoBaP. For SP, IHT, BP and EMBGAMP, we use the implementations available on author's webpages (resp. at *http://sites.google.com/site/igorcarron2/cscodes/*, *http://www.personal.soton.ac.uk/tb1m08/sparsify/sparsify.html*, *http://www.acm.caltech.edu/l1magic/* (ℓ_1-magic) and *http://www2.ece.ohio-state.edu/vilaj/EMBGAMP/EMBGAMP.html*). MP is run until the ℓ_2-norm of the residual drops below $\sqrt{N\sigma_n^2}$. The same criterion is used for BP. BMP iterates as long as $\log p(\mathbf{y}, \hat{\mathbf{x}}^{(n)}, \hat{\mathbf{s}}^{(n)}) > \log p(\mathbf{y}, \hat{\mathbf{x}}^{(n-1)}, \hat{\mathbf{s}}^{(n-1)})$ where n is the iteration number (see [32]). SoBaP is run until $\forall i \in \{1, \ldots, M\}$, $|q(s_i^{(n)}) - q(s_i^{(n-1)})| < 10^{-2}$. Finally, we set $p_i = \frac{K}{M}$, $\forall i$.

1) Gaussian model: In this scenario, the amplitudes of the non-zero coefficients are drawn from a Gaussian distribution, according to (6). We set $\sigma_{x_i}^2 = 1 \; \forall i$.

Fig. 1(a) shows the MSE as a function of the number of non-zero coefficients, K. For $K \leq 40$, SoBaP presents, together with EMBGAMP, the best performance. Beyond this bound, it is dominated by EMBGAMP but keeps a good behaviour with regard to other algorithms.

Fig. 1(b) and (c) represent the algorithm performance in terms of the reconstruction of the SR support. We can observe that SoBaP succeeds in keeping both reasonable missed detection and false detection rates on a large range of sparsity levels. This is not the case for the other algorithms. If some of them present better performance for one rate, this is at the expense of the other one. BP and EMBGAMP constitute two extreme examples. They are both based on a "spender" strategy: they prefer missing no atom (their missed detection is equal to zero, that is why they do not appear in Fig. 1(b)) even if they are not sure they are good ones.

It is difficult to compare the running times of the considered algorithms since they do not have the same stopping criteria. In Fig. 5, we see that SoBaP presents a computational cost higher than MP or BMP, while sharing a similar complexity order per update step (see section IV-B). This can be explained by the fact that SoBaP updates all indices at each iteration, as we previously mentioned. Beyond these observations, SoBaP remains competitive with the other algorithms, in particular for high sparsities (*i.e.,* small numbers of non-zero coefficients). Note finally that EMBGAMP constitutes here the most costly procedure, with a high constant running time.

*2) "*0-1*" model:* In this second scenario, the amplitude of the non-zero coefficients in $\mathbf{x}$ are forced to be equal to 1.

Fig. 2(a) shows the MSE as a function of the number of non-zero coefficients, K. For this particular setup, we experimentally observed that SoBaP presents better results when we set $\sigma_{x_i}^2 = 0.1 \ \forall i$ in the algorithm. Thus, SoBaP outperforms all algorithms (or present similar performance, for high sparsities) except EMBGAMP which clearly dominates.

The performance achieved in terms of reconstruction of the SR support (see Fig. 2(b) and (c)) is similar to the one observed in the previous scenario. SoBaP constitutes the best compromise between missed and false detection rates, while BP and EMBGAMP follow the same strategy as before: they select all atoms, including "bad" ones (*i.e.,* not used to generate the data).

B. Structured case

In the structured case, the links between atoms are taken into account in the sparse decomposition. For the experiments, we considered two different structures: a Markov chain, for which we showed the equivalence with a particular Boltzmann machine in (11)-(12), and a general, non-dedicated Boltzmann machine.

1) Markov chain: We first consider the simple scenario where the positions of the non-zero coefficients in $\mathbf{x}$ follow a Markov-Chain model. We use the following parameters: $N = 128$, $M = 256$, $\sigma_n^2 = 10^{-3}$. The observations are generated as follows. The elements of the dictionary are drawn, for each observation, from a zero-mean Gaussian distribution with variance N^{-1}. For each point of simulation, we fix the number of non-zero coefficients and select their positions uniformly at random. Either the Gaussian or the "0-1" model is considered for the amplitudes of the non-zero coefficients in $\mathbf{x}$.

We consider the case of a symetric Markov chain *i.e.,* $p_{i+1}^0 = p_{i+1}^1 \ \forall i$. The value of the p_{i+1}^0's are drawn as follows

$$p_{i+1}^0 \sim \mathcal{U}[0, 0.5] \quad \text{if } s_i = s_{i+1}, \tag{54}$$

$$p_{i+1}^0 \sim \mathcal{U}[0.5, 1] \quad \text{otherwise.} \tag{55}$$

Boltzmann parameters $\mathbf{b}$ and $\mathbf{W}$ are then constructed according to (11)-(12). The performance of SSoBaP is represented in Fig. 3 for both the Gaussian (dashed curves) and "0-1" (solid curves) models. SSoBaP is compared to the greedy procedure proposed by Faktor *et al.* in [54], called BM_MAP_OMP. The latter also relies on a Boltzmann machine. The same parameters are thus used in both algorithms. The unstructured variant of SSoBaP, SoBaP, is considered too, in order to assess the relevance of accounting sparse structures with the BM parameters. BM_MAP_OMP iterates until the ℓ_2-norm of the residual drops below

10^{-3} or the iteration number exceeds $N/2$. SoBaP and SSoBaP are run until $|q(s_i^{(n)}) - q(s_i^{(n-1)})| < 10^{-2}$ $\forall i \in \{1, \ldots, M\}$.

We see in Fig. 3 that SSoBaP nicely takes benefit from the additional information on the SR support and thus improves the performance of SoBaP with respect to all figures of merit. In the Gaussian case, the probability of missed detection is roughly equal to 5.10^{-2} (versus 10^{-1} for SoBaP) over a large range of sparsity levels with a probability of false detection of about 10^{-3}. In the "0-1" model, no missed detections have been detected up to $K = 58$ and the probability of false detection is of the order of 10^{-3} in this range. These good properties in terms of support recovery are confirmed in Fig 3(a) by the MSE performance. Let us note, that for this particular scenario, BM_MAP_OMP exhibits the worst performance. In particular, its probability of false detection rapidly increases as the number of non-zero coefficients becomes larger.

Finally, Fig. 6 illustrates the running time of the three procedures. We note that the computational burden induced by SoBaP strongly depends on the considered scenario. On the other hand, the running times of SSoBaP and BM_MAP_OMP remain similar for both the Gaussian and "0-1" models. As far as this simulation setup is concerned, SSoBaP is significantly faster than BM_MAP_OMP for small to moderate values of K.

2) "General" Boltzmann machine: We consider the following parameters: $N = 32$, $M = 64$, $\sigma_n^2 = 10^{-3}$ and generate the data as follows. The elements of the dictionary are drawn, for each observation, from a zero-mean Gaussian with variance N^{-1}. For each point of simulation, we fix the number of non-zero coefficients *and* their positions in the SR support. These positions are thus drawn uniformly at random *once* for *all* observations. This leads to a particular support $\mathbf{s}$ that we use for all trials. So, we average the performance of the algorithms on data structured in the same way. Regarding the amplitudes of the non-zero coefficients in $\mathbf{x}$, we consider the same scenarios as for the unstructured case, *i.e.,* the Gaussian and "0-1" models.

The parameters of the Boltzmann machine, $\mathbf{b}$ and $\mathbf{W}$, are drawn from the a posteriori distribution $p(\mathbf{b}, \mathbf{W}|\mathbf{s})$ by means of the "Single-variable Exchange" algorithm introduced in [67], using $w_{ij} \sim \mathcal{U}[-1, 1]$ $\forall i, j$ and $b_i \sim \mathcal{U}[-20, 20]$ $\forall i$ as a priori distributions. We initialize all elements in $\mathbf{b}$ and $\mathbf{W}$ to 0. For each point of simulation, the "Single-variable Exchange" algorithm is run with a burn-out iteration number of 1000; we then allocate the 500 following parameter realizations to the 500 observations of the considered point.

SSoBaP is compared to BM_MAP_OMP and SoBaP. BM_MAP_OMP iterates until the ℓ_2-norm of

the residual drops below 10^{-3} or the iteration number exceeds $N/2$. SoBaP and SSoBaP are run until $|q(s_i^{(n)}) - q(s_i^{(n-1)})| < 10^{-2}\ \forall i \in \{1, \ldots, M\}$.

Fig. 4(a), (b) and (c) sums up the performance achieved by the three algorithms under the two considered scenarios.

Focusing on the Gaussian model (dashed curves), we observe that SSoBaP dominates SoBaP and BM_MAP_OMP in terms of MSE for a wide range of sparsity levels. Moreover, it presents stable missed and false detection rates (around 10^{-2}). We then can see that it outperforms SoBaP and BM_MAP_OMP in terms of missed detection rate for all considered sparsities while achieving the lowest false detection rate for small sparsities ($K > 10$).

SSoBaP keeps its general good behaviour with the "0-1" model (solid curves). This good behaviour is even reinforced by zero missed detection for $K < 13$. Note that this does not contradict the similarity observed between the MSE curves: missed and false detection rates impact on the MSE but their influence is difficult to measure, as a high MSE can be due to high missed and false detection rates but also to a bad coefficients' estimation.

Fig. 7 shows the running times of the considered algorithms in both the Gaussian and "0-1" scenarios. As far as these setups are concerned, SSoBaP has always a smaller running time than BM_MAP_OMP. The behaviour of SoBaP differs according to the considered scenario. For the Gaussian model (dashed curves), SoBaP has the smallest running time among the three algorithms. For the "0-1" model (solid curves), SoBaP is outperformed by SSoBaP.

VI. 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 then focus on the resolution of marginalized MAP problems. The proposed approach is based on a mean-field approximation and the use of the "variational Bayes Expectation-Maximization" algorithm, and results in the so-called "Structured Soft Bayesian Pursuit" (SSoBaP) algorithm. We assess the performance of SSoBaP in the unstructured and structured cases (the unstructured version of SSoBaP is then called SoBaP). In both cases, we evaluate the ability of the algorithm to reconstruct the SR support and estimate the non-zero coefficients. Experimental results show that the corresponding algorithms perform well in comparison to other state-of-the-art algorithms, at a reasonable computational cost.

Future work will consider the use of the proposed algorithm in practical applications, in particular in audio processing where structured sparsity can be favourably exploited for effcient representations of

audio signals.

VII. Acknowledgements

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

References

[1] L. Daudet, "Sparse and structured decompositions of audio signals in overcomplete spaces," in *Proc. Int'l Conference on Digital Audio Effects (DAFx)*, Naples, Italy, October 2004, pp. 22 – 26.

[2] C. Févotte, L. Daudet, S. J. Godsill, and B. Torrésani, "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.

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

[4] R. M. Figueras i Ventura, P. Vandergheynst, and P. Frossard, "Low rate and scalable image coding with redundant representations," Tech. Rep., TR-ITS-03.02, June 2003.

[5] D. L. Donoho, "Compressed sensing," *IEEE Trans. On Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[6] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal of Computing*, vol. 24, no. 2, pp. 227–234, April 1995.

[7] A. Miller, *Subset selection in regression*, Chapman and Hall/CRC, 2nd edition, April 2002.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407499, 2004.

[9] H. Markowitz, "The optimization of a quadratic function subject to linear constraints," *Naval Research Logistics Quaterly*, vol. 3, no. 1-12, pp. 111–133, March-June 1956.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, pp. 267–288, 1996.

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

[13] N. G. Kingsbury and T. H. Reeves, "Overcomplete image coding using iterative projection-based noise shaping," in *Proc. IEEE Int'l Conference on Image Processing (ICIP)*, 2002, vol. 3, pp. 597–600.

[14] 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, December 2008.

[15] I. Daubechies, M. Defrise, and C. DeMol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

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

[17] Y. C. Pati, R. Rezaiifar, 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.

[18] C.-T. Chen, "Adaptive transform coding via quadtree-based variable blocksize dct," in *Proc. IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 23-26 May 1989.

[19] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," Tech. Rep., Stanford University, March 2006.

[20] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. On Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[21] D. Needell and J. A. Tropp, "Cosamp: iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009.

[22] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by v1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[23] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Journal of Neural Computation*, vol. 12, pp. 337–365, 2000.

[24] M. Girolami, "A variational method for learning sparse and overcomplete representation," *Journal of Neural Computation*, vol. 13, no. 11, pp. 2517–2532, 2003.

[25] C. Févotte and S. J. Godsill, "A bayesian approach for blind separation of sparse sources," *IEEE Trans. On Acoustics, Speech and Signal Processing*, vol. 14, no. 6, pp. 2174–2188, November 2006.

[26] C. Févotte and S. J. Godsill, "Blind separation of sparse sources using jeffrey's inverse prior and the expectation-maximization algorithm," in *Proc. Int'l Conference on Independent Component Analysis and Blind Source Separation (ICA)*, 2006, pp. 593–600.

[27] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Proc. Workshop on Information Theory and Applications (ITA)*, La Jolla, CA, January 2008, pp. 326 – 333.

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

[29] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "An iterative bayesian algorithm for sparse component analysis in presence of noise," *IEEE Trans. On Signal Processing*, vol. 57, no. 11, pp. 4378–4390, November 2009.

[30] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," Tech. Rep., available at http://arxiv.org: 0812.4627v2.pdf, June 2009.

[31] C. Herzet and A. Drémeau, "Sparse representation algorithms based on mean-field approximations," in *Proc. IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, USA, March 2010, pp. 2034–2037.

[32] C. Herzet and A. Drémeau, "Bayesian pursuit algorithms," in *Proc. European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, August 2010.

[33] C. Soussen, J. Idier, D. Brie, and J. Duan, "From bernoulli-gaussian deconvolution to sparse signal restoration," Tech. Rep., CRAN/IRCCyN, January 2010.

[34] D. Ge, J. Idier, and E. Le Carpentier, "Enhanced sampling schemes for mcmc based blind bernoulli-gaussian deconvolution," in *Signal Processing*, April 2011, vol. 91, pp. 759 – 772.

[35] F. Krzakala, M. Mézart, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical physics-based reconstruction in compressed sensing," available on http://arxiv.org/abs/1109.4424v2, 2011.

[36] J. Vila and P. Schniter, "Expectation-maximization bernoulli-gaussian approximate message passing," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, November 2011.

[37] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. On Information Theory*, vol. 55, no. 11, pp. 5302–5316, November 2009.

[38] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Compressed sensing of block-sparse signals: uncertainty relations and efficient recovery," Submitted to IEEE Trans. On Signal Processing, 2010.

[39] 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.

[40] L. Yu, J.-P. Barbot, G. Zheng, and H. Sun, "Compressive sensing for cluster structured sparse signals: variational bayes approach," submitted to IEEE Trans. On Signal Processing, 2011.

[41] L. Yu, H. Sun, J.-P. Barbot, and G. Zheng, "Bayesian compressive sensing for clustered sparse signals," in *Proc. IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 3948 – 3951.

[42] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," in *Proc. Int'l Conference on Machine Learning*, 2009.

[43] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, "Hybrid approximate message passing with applications to structured sparsity," available on http://arxiv.org/abs/1111.2581, 2011.

[44] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. IEEE Information Theory Workshop (ITW)*, Cairo, Egypt, January 2010, pp. 1–5.

[45] P. Sprechmann, I. Ramirez, and G. Sapiro, "Collaborative hierarchical sparse modeling," in *Proc. IEEE Int'l Conference on Information Sciences and Systems (CISS)*, March 2010.

[46] M. Kowalski and B. Torrésani, "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.

[47] L. Daudet, "Sparse and structured decompositions of signals with the molecular matching pursuit," *IEEE Trans. On Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1808 – 1816, September 2006.

[48] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," Tech. Rep., INRIA, 2010.

[49] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Trans. On Signal Processing*, vol. 57, no. 9, pp. 3488–3497, September 2009.

[50] L. He, H. Chen, and L. Carin, "Tree-structured compressive sensing with variational bayesian analysis," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 233–236, 2010.

[51] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. IEEE Annual conference on Information Sciences and Systems (CISS)*, Princeton, NJ, March 2010, pp. 1–6.

[52] 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.

[53] 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.

[54] T. Faktor, Y. C. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," Submitted to IEEE Trans. On Signal Processing.

[55] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge, "Model-based compressive sensing," *IEEE Trans. On Information Theory*, vol. 56, pp. 1982–2001, April 2010.

[56] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[57] Bernard C. Levy, *Principles of Signal Detection and Parameter Estimation*, Springer, 1 edition, July 2008.

[58] M. J. Wainwright and M. I. Jordan, "Graphical models, variational inference and exponential families," Tech. Rep., UC Berkeley, Dept. of Statistics, 2003.

[59] M. Beal, *Variational algorithms for approximate Bayesian inference*, Ph.D. thesis, University College of London, May 2003.

[60] T. P. Minka, "Using lower bounds to approximate integrals," 2001.

[61] 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, pp. 453–463, 2003.

[62] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[63] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[64] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," *Learning in graphical models*, vol. 89, pp. 355 – 368, 1998.

[65] A. Drémeau, C. Herzet, and L. Daudet, "Soft bayesian pursuit algorithm for sparse representations," in *Proc. IEEE Int'l Statistical Signal Processing Workshop (SSP)*, 2011, pp. 341–344.

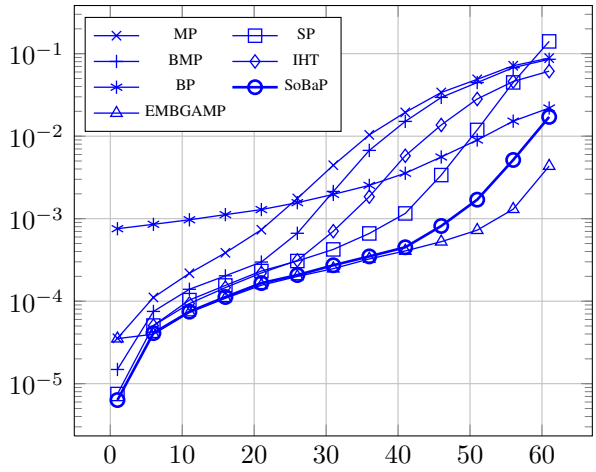
[66] H. Zayyani, M. Babaie-Zadeh, and C. Jutten, "Sparse component analysis in presence of noise using em-map," in *Proc. Int'l Conf. on Independent Component Analysis and Signal Separation*, London, 2007.

[67] 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.

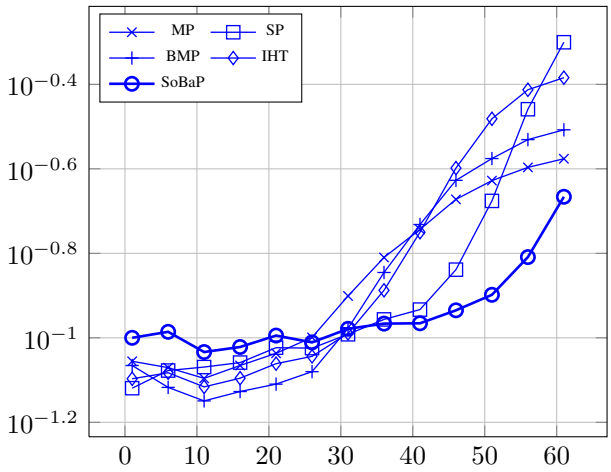
[68] M. J. Nijman and H. J. Kappen, "Efficient learning in sparsely connected boltzmann machines," in *Proc. Int'l Conf. on Artificial Neural Networks*, 1996.

[69] N. L. Lawrence, C. M. Bishop, and M. I. Jordan, "Mixture representations for inference and learning in boltzmann machines," in *Proc. Conference on Uncertainty in Artificial Intelligence*, 1998.

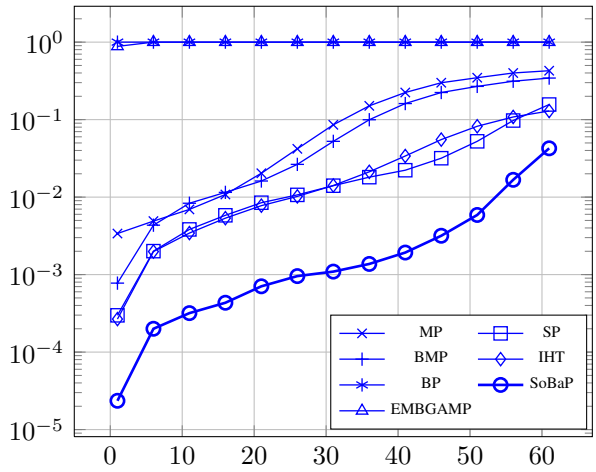
[70] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," available on http://arxiv.org/abs/1010.5141, 2010.



(a) MSE on the coefficients

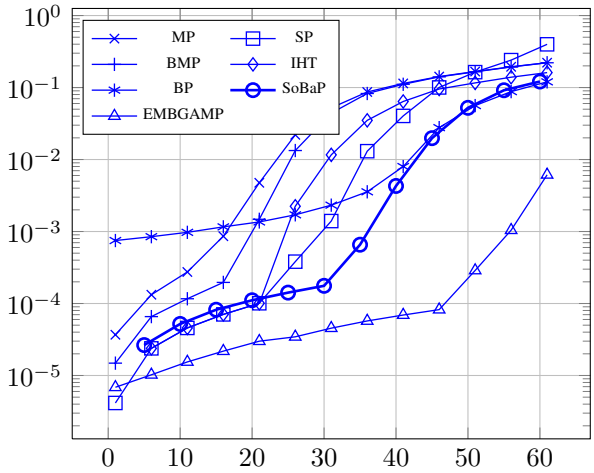


(b) Probability of missed detection

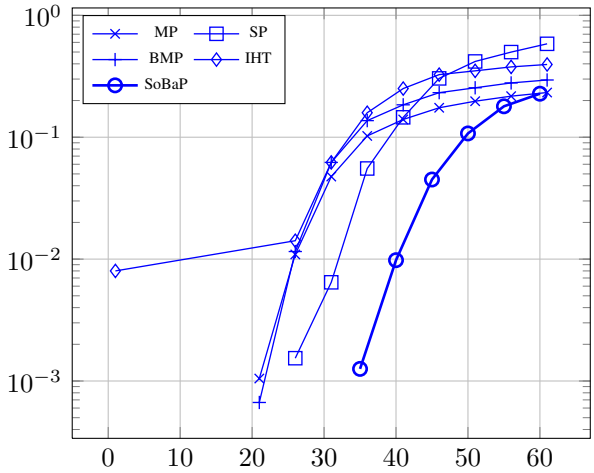


(c) Probability of false detection

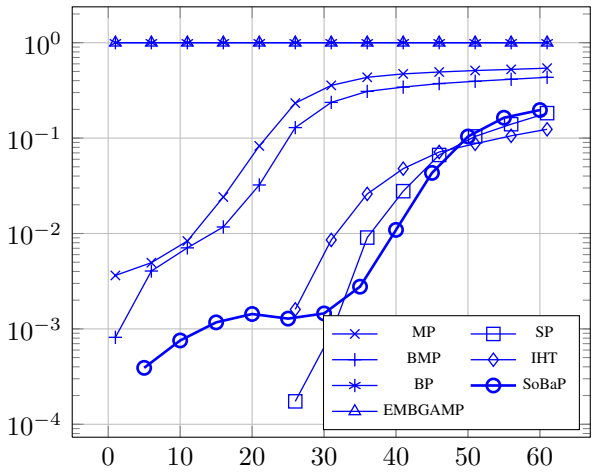
Figure 1. MSE (a), probability of missed (b) and false (c) detection versus K. The support of the sparse vector is drawn uniformaly at random. The non-zero coefficients in $\mathbf{x}$ follow the "Gaussian model" and $\sigma_n^2 = 10^{-3}$.



(a) MSE on the coefficients

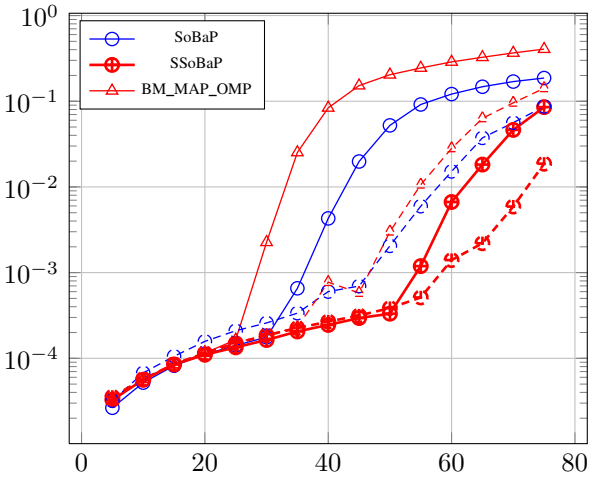


(b) Probability of missed detection

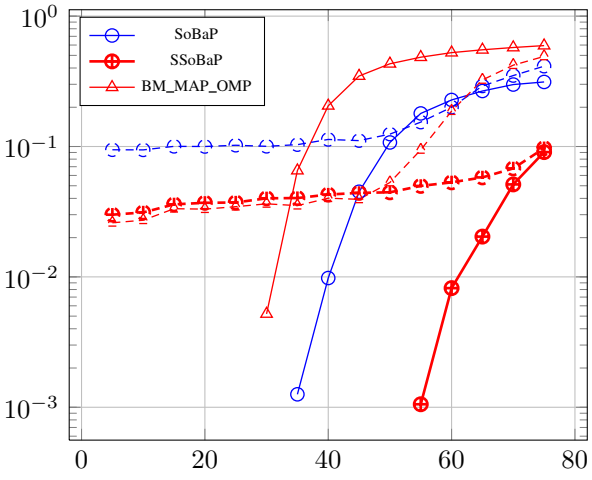


(c) Probability of false detection

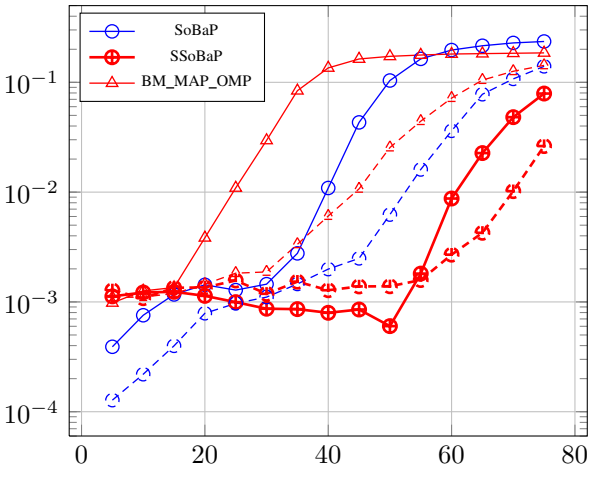
Figure 2. MSE (a), probability of missed (b) and false (c) detection versus K. The support of the sparse vector is drawn uniformaly at random. The non-zero coefficients in $\mathbf{x}$ follow the "0-1 model" and $\sigma_n^2 = 10^{-3}$.



(a) MSE on the coefficients

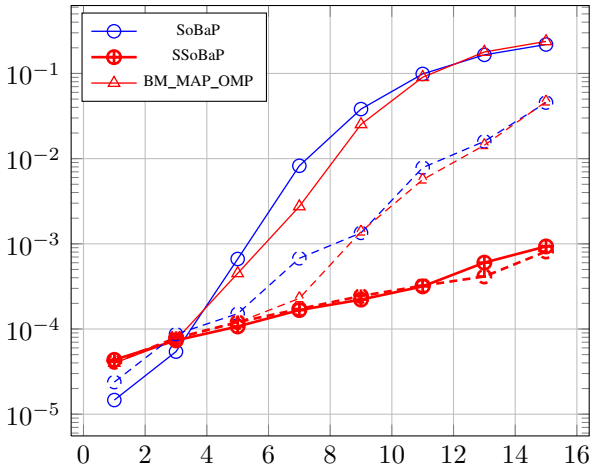


(b) Probability of missed detection

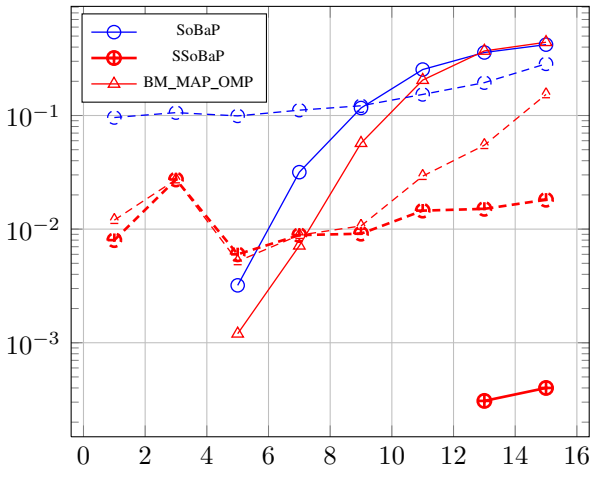


(c) Probability of false detection

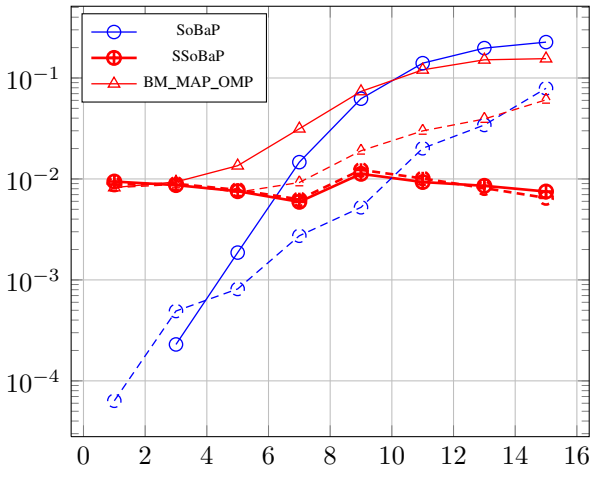
Figure 3. MSE (a), probability of missed (b) and false (c) detection versus K. The support of the sparse vector follows a Markov chain model. The non-zero coefficients in $\mathbf{x}$ follow the "0-1" (solid) or "Gaussian" (dashed) model and $\sigma_n^2 = 10^{-3}$.



(a) MSE on the coefficients



(b) Probability of missed detection



(c) Probability of false detection

Figure 4. MSE (a), probability of missed (b) and false (c) detection versus K. The support of the sparse vector follows the general model of a Boltzmann machine. The non-zero coefficients in $\mathbf{x}$ follow the "0-1" (solid) or "Gaussian" (dashed) model and $\sigma_n^2 = 10^{-3}$.

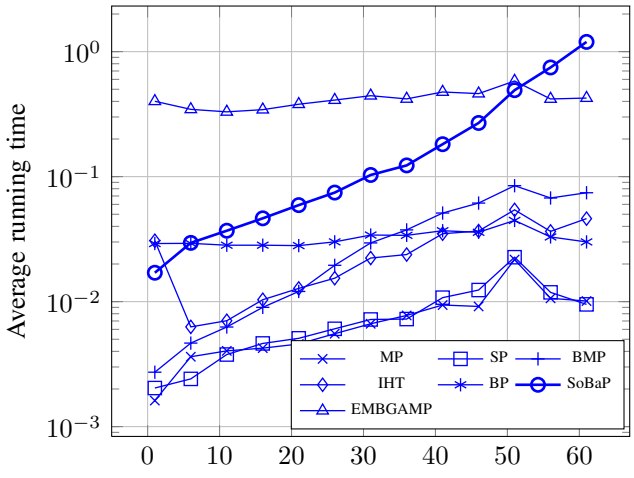


Figure 5. Average running time versus K. The support of the sparse vector is drawn uniformaly at random. The non-zero coefficients in $\mathbf{x}$ follow the "Gaussian model" and $\sigma_n^2 = 10^{-3}$.

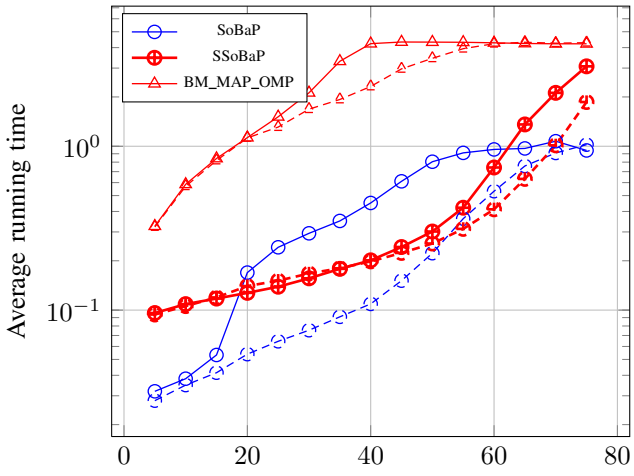


Figure 6. Average running time versus K. The support of the sparse vector follows the Markov chain model. The non-zero coefficients in $\mathbf{x}$ follow the "0-1" (solid) or "Gaussian" (dashed) model and $\sigma_n^2 = 10^{-3}$.

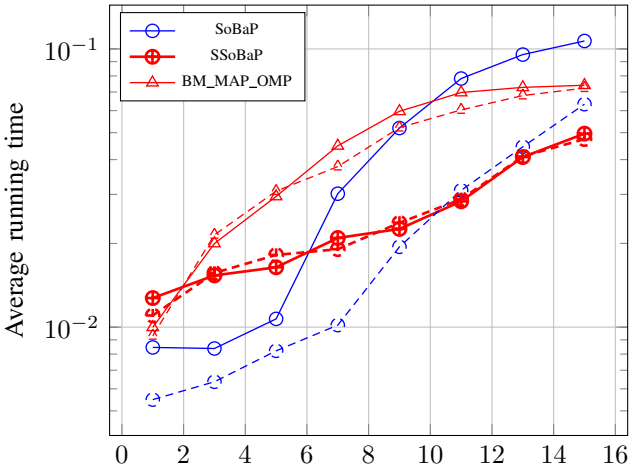


Figure 7. Average running time versus K. The support of the sparse vector follows the general model of a Boltzmann machine. The support of the sparse vector follows the general model of a Boltzmann machine. The non-zero coefficients in $\mathbf{x}$ follow the "0-1" (solid) or "Gaussian" (dashed) model and $\sigma_n^2 = 10^{-3}$.