SPARSE REPRESENTATION ALGORITHMS BASED ON MEAN-FIELD APPROXIMATIONS

C. Herzet and A. Drémeau

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

ABSTRACT

In this paper we address the problem of sparse representation (SR) within a Bayesian framework. We assume that the observations are generated from a Bernoulli-Gaussian process and consider the corresponding Bayesian inference problem. Tractable solutions are then proposed based on the "mean-field" approximation and the variational Bayes EM algorithm. The resulting SR algorithms are shown to have a tractable complexity and very good performance over a wide range of sparsity levels. In particular, they significantly improve the critical sparsity upon state-of-the-art SR algorithms.

***Index Terms*—** Sparse representations, Bayesian framework, variational methods, mean-field approximation.

1. INTRODUCTION

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

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

where $\|\cdot\|_p$ denotes the l_p-norm[1].

Finding the exact solution of (1) is an intractable problem. Therefore, numerous suboptimal (but tractable) algorithms have been devised in the literature to address the SR problem. We can roughly divide the existing algorithms into 3 main families: *i) the greedy algorithms* which build up the sparse vector $\mathbf{x}$ by making a succession of locally-optimal decisions. This family includes MP [1], OMP [2], gradient pursuit (GP) [3] , CoSamp [4] and subspace pursuit (SP) [5] algorithms; *ii) the algorithms based on a problem relaxation*, like basis pursuit (BP) [6], FOCUSS [7] or SL0 [8]; these algorithms approximate (1) by relaxed problems that can be solved efficiently by standard optimization procedures; *iii) the Bayesian algorithms* which express the sparse representation problem as the solution of a Bayesian inference problem and apply statistical tools to solve it. Examples of such algorithms include the relevant vector machine (RVM) algorithm [9], the sum-product [10] and the expectation-maximization [11] SR algorithms.

In this paper we place the SR problem into a Bayesian framework: the observed vector $\mathbf{y}$ is modeled as the output of a Bernoulli-Gaussian (BG) process and the sparse vector $\mathbf{x}$ is searched as the solution of the corresponding Bayesian inference problem. Tractable solutions of this problem are computed by considering mean-field variational approximations of $p(\mathbf{x}|\mathbf{y})$. The implementation of the MF approximations is made by means of the variational Bayes EM (VB-EM) algorithm. The resulting SR algorithms are shown to have a tractable complexity and, as far as our simulation setup is concerned, lead to a significant improvement of the performance upon state-of-the-art SR algorithms.

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

2. BG FORMULATION OF THE SR PROBLEM

In this section, we present the BG probabilistic model which will be used in section 3 to derive SR algorithms.

We assume that the observed vector $\mathbf{y}$ has a Gaussian distribution with mean $\mathbf{D}\mathbf{x}$ and covariance $\sigma_n^2 \mathbf{I}_N$, *i.e.*,

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

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. We suppose moreover that $\mathbf{x}$ obeys the following probabilistic model:

$$p(\mathbf{x}) = \sum_{\mathbf{s}} p(\mathbf{x}, \mathbf{s}), \tag{3}$$

$$= \prod_{i=1}^{M} \sum_{s_i} p(x_i|s_i) p(s_i), \tag{4}$$

where

$$p(x_i|s_i) = \mathcal{N}(0, \sigma^2(s_i)), \tag{5}$$

$$p(s_i) = \mathrm{Ber}(p_i), \tag{6}$$

and $\mathrm{Ber}(p_i)$ denotes a Bernoulli distribution with parameter p_i. The probability $p(\mathbf{x})$ can be interpreted as follows: each component x_i is drawn, independently, from a mixture of two zero-mean Gaussians whose variance depends on the realization of a Bernoulli variable s_i.

The BG model (3)-(6) is actually well-suited to modeling situations where $\mathbf{x}$ is sparse. Indeed, if $\sigma^2(s_i = 0) \ll \sigma^2(s_i = 1)$ and $p_i \ll 1 \; \forall \, i$, most of the components x_i will be drawn from a Gaussian with a small variance. Hence, only a small fraction of the x_i's will have an amplitude significantly larger than the others. This is clearly in accordance with the sparse representation paradigm.

Model (3)-(6) (or variants thereof) has already been used in many Bayesian algorithms available in the literature, see *e.g.*, [10, 11, 12]. In this paper, we present a new approach, based on mean-field variational approximations, to solve SR problems.

3. SR ALGORITHMS BASED ON MEAN-FIELD VARIATIONAL APPROXIMATIONS

Based on probabilistic model (3)-(6), sparse solutions for $\mathbf{x}$ can be found as the maximum or the mean of posterior distribution $p(\mathbf{x}|\mathbf{y})$:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \; \log p(\mathbf{x}|\mathbf{y}), \tag{7}$$

$$\hat{\mathbf{x}} = \int \mathbf{x} \, p(\mathbf{x}|\mathbf{y}) d\mathbf{x}, \tag{8}$$

where

$$p(\mathbf{x}|\mathbf{y}) = \sum_{\mathbf{s}} p(\mathbf{x}, \mathbf{s}|\mathbf{y}). \tag{9}$$

The complexity associated to problem (7)-(8) is intractable. In particular, the marginalization in (9) requires a number of operations which scales exponentially with the dimension of $\mathbf{s}$.

Dealing with complex marginalizations is a common issue in statistics. One possible way to solve this kind of problem is to resort to variational approximations of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$ (see *e.g.*, [13] for a survey). The simplest (and probably the most common) variational approximation is the so-called "mean-field" (MF) approximation, which forces the independence between (some of) the variables. In this paper, we propose to apply the mean-field approximation to the SR problem.

We recall the basics of mean-field approximations in section 3.1. Then, in sections 3.2 to 3.4 we derive SR algorithms based on different mean-field approximations of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$.

3.1. Mean-field approximation: Basics

Let $\boldsymbol{\theta}$ denote a vector of random variables (*e.g.*, $\boldsymbol{\theta} = [\mathbf{x}^T \mathbf{s}^T]^T$) and let $p(\boldsymbol{\theta}|\mathbf{y})$ be its a posteriori probability. The mean-field approximation of $p(\boldsymbol{\theta}|\mathbf{y})$ approximates $p(\boldsymbol{\theta}|\mathbf{y})$ by a probability having a "suitable" factorization. More precisely, let $q(\boldsymbol{\theta})$ be a probability distribution such that[2]:

$$q(\boldsymbol{\theta}) \triangleq q(\boldsymbol{\theta}_1)\, q(\boldsymbol{\theta}_2), \tag{10}$$

where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_1$ are such that $\boldsymbol{\theta}^T = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T]$ and $\int q(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i = 1$. Then, the mean-field approximation of $p(\boldsymbol{\theta}|\mathbf{y})$, say $q^\star(\boldsymbol{\theta})$, can be expressed as

$$q^\star(\boldsymbol{\theta}) = \arg\min_{q(\boldsymbol{\theta})} \mathrm{KL}\big(q(\boldsymbol{\theta}); p(\boldsymbol{\theta}|\mathbf{y})\big), \tag{11}$$
$$\textit{subject to } (10),$$

where $\mathrm{KL}\big(q(\boldsymbol{\theta}); p(\boldsymbol{\theta}|\mathbf{y})\big)$ is the Kullback-Leibler distance between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{y})$, *i.e.*,

$$\mathrm{KL}\big(q(\boldsymbol{\theta}); p(\boldsymbol{\theta}|\mathbf{y})\big) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})} d\boldsymbol{\theta}. \tag{12}$$

The solution of optimization problem (11) can be iteratively computed via the following recursion:

$$q^{(n+1)}(\boldsymbol{\theta}_1) \propto \exp\left\{ \langle \log p(\mathbf{y}, \boldsymbol{\theta}) \rangle_{q^{(n)}(\boldsymbol{\theta}_2)} \right\}, \tag{13}$$

$$q^{(n+1)}(\boldsymbol{\theta}_2) \propto \exp\left\{ \langle \log p(\mathbf{y}, \boldsymbol{\theta}) \rangle_{q^{(n+1)}(\boldsymbol{\theta}_1)} \right\}, \tag{14}$$

where $\propto$ denotes equality up to a normalization factor and

$$\langle \log p(\mathbf{y}, \boldsymbol{\theta}) \rangle_{q(\boldsymbol{\theta}_i)} \triangleq \int q(\boldsymbol{\theta}_i) \log p(\mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta}_i. \tag{15}$$

The algorithm defined by (13)-(14) is usually referred to as *"variational Bayes EM (VB-EM) algorithm"* in the literature [14]. This algorithm is ensured to converge to a saddle point or a (local or global) maximum of problem (11).

[2]For the sake of conciseness, we limit here our discussion to the case where $q(\boldsymbol{\theta})$ is constrained to have a factorization as the product of *two* factors. The extension to the general case is straightforward.

Note that the MF approximation automatically provides an approximation of the marginals of $p(\boldsymbol{\theta}|\mathbf{y})$ with respect to $\boldsymbol{\theta}_i$'s. Indeed, we have for $i \neq j$:

$$\begin{aligned} p(\boldsymbol{\theta}_i|\mathbf{y}) &= \int p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}_j, \\ &\simeq \int q^\star(\boldsymbol{\theta}) d\boldsymbol{\theta}_j = \int q^\star(\boldsymbol{\theta}_i) q^\star(\boldsymbol{\theta}_j) d\boldsymbol{\theta}_j, \\ &= q^\star(\boldsymbol{\theta}_i), \end{aligned} \tag{16}$$

where the last equality follows from the fact that $q(\boldsymbol{\theta}_i)'s$ are constrained to be probability distributions (*i.e.*, $\int_{\boldsymbol{\theta}_i} q(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i = 1 \; \forall i$). Hence, the VB-EM algorithm can also be regarded as an iterative procedure which computes approximations of the marginals of $p(\boldsymbol{\theta}|\mathbf{y})$.

3.2. MF approximation $p(\mathbf{x}, \mathbf{s}|\mathbf{y}) \simeq q(\mathbf{x})q(\mathbf{s})$

In this section, we particularize the VB-EM equations (13)-(14) to the case where the MF approximation of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$ is constrained to have the following structure:

$$q(\mathbf{x}, \mathbf{s}) = q(\mathbf{x})q(\mathbf{s}). \tag{17}$$

Let $\Sigma_{\mathbf{s}}$ be a diagonal matrix whose ith diagonal element is defined as $(\Sigma_{\mathbf{s}})_{ii} \triangleq \sigma^2(s_i)$. Then, taking (2)-(6) into account, we obtain[3] after some mathematical manipulations:

$$\begin{aligned} q(\mathbf{s}) &\propto (\sqrt{\det \Sigma_{\mathbf{s}}})^{-1} \exp\left\{ -\frac{1}{2} \langle \mathbf{x}^T \Sigma_{\mathbf{s}}^{-1} \mathbf{x} \rangle_{q(\mathbf{x})} \right\} p(\mathbf{s}), \\ &\propto \prod_i \frac{1}{\sqrt{\sigma^2(s_i)}} \exp\left\{ -\frac{\langle x_i^2 \rangle_{q(x_i)}}{2\sigma^2(s_i)} \right\} p(s_i), \end{aligned} \tag{18}$$

$$q(\mathbf{x}) = \mathcal{N}(m, \Gamma), \tag{19}$$

where

$$\Gamma = \left(\frac{\mathbf{D}^T \mathbf{D}}{\sigma_n^2} + \langle \Sigma_{\mathbf{s}}^{-1} \rangle_{q(\mathbf{s})} \right)^{-1}, \tag{20}$$

$$m = \frac{1}{\sigma_n^2} \Gamma \mathbf{D}^T \mathbf{y}. \tag{21}$$

As mentioned in section 3.1, $q(\mathbf{x})$ can also be regarded as an approximation of $p(\mathbf{x}|\mathbf{y})$. Coming back to MAP problem (7), we then have

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\max_{\mathbf{x}} \; \log p(\mathbf{x}|\mathbf{y}), \\ &\simeq \arg\max_{\mathbf{x}} \log q(\mathbf{x}) = m. \end{aligned}$$

Moreover, since $q(\mathbf{x})$ is Gaussian, m is also the solution of (8). After convergence of the VB-EM algorithm, m is therefore an approximation of the sparse solutions (7)-(8).

In the sequel, we will refer to the procedure defined in (18)-(21) as variational Bayes sparse representation (VBSR1) algorithm. The complexity of VBSR1 is dominated by the matrix inversion in (20). This operation can be performed with a complexity scaling as N^3 by using the Matrix Inversion Lemma [15]. This order of complexity is similar to the one of algorithms based on problem relaxation such as BP or FOCUSS.

[3]When clear from the context, we will drop the iteration indices $^{(n)}$ in the rest of the paper.

It is quite interesting to compare VBSR1 with CoSaMP/SP algorithms [4]-[5]. At each iteration, CoSaMP and SP select a subset of atoms and compute the least-square (LS) estimate of $\mathbf{x}$ assuming these atoms are active. The subset of active atoms is selected on the basis of the amplitude of the scalar products between the atoms and the current residual. VBSR1 performs similar operations but introduces "softness" in both the selection of the active atoms and the computation of the estimate of $\mathbf{x}$: *i)* instead of making hard decisions about the atom activity, VBSR1 rather computes a *probability* of activity for each atom, see (18); *ii)* VBSR1 computes the linear minimum mean square estimate (LMMSE) of $\mathbf{x}$ assuming that $\mathbf{x}$ is zero-mean with covariance matrix $\langle \Sigma_{\mathbf{s}} \rangle_{q(\mathbf{s})}$, see (21). It can be seen that the LMMSE and the LS estimates are equal when all the diagonal elements of $\langle \Sigma_{\mathbf{s}} \rangle_{q(\mathbf{s})}$ are either equal to 0 or ∞. This situation occurs when $q(\mathbf{s})$ is a Dirac function, *i.e.*, when hard decisions are made about the atom activity.

VBSR1 also shares some similarities with RVM since both algorithms compute a new Gaussian density on $\mathbf{x}$ at each iteration. However, the two algorithms make very different hypotheses about the prior model on $\mathbf{x}$. In RVM, $p(x_i)$ is assumed to be Gaussian with unknown variance; a new estimate of the variance is then recomputed at each iteration. On the other hand, VBSR1 is based on a BG model on $p(x_i)$. At each iteration, a new approximation $q(\mathbf{s})$ is computed. Hence, as shown in section 4, RVM and VBSR1 lead to very different performance in practice.

3.3. MF Approximation $p(\mathbf{x}, \mathbf{s}|\mathbf{y}) \simeq \prod_i q(x_i, s_i)$

In this section, we consider the MF approximation of $p(\mathbf{x}, \mathbf{s}|\mathbf{y})$ when $q(\mathbf{x}, \mathbf{s})$ is constrained to have the following factorization[4]:

$$q(\mathbf{x}, \mathbf{s}) = \prod_{i=1}^{M} q(x_i, s_i). \tag{22}$$

Particularizing (18)-(19) to the BG model (2)-(6), we obtain after some manipulations:

$$q(x_i, s_i) = q(x_i|s_i)q(s_i) \qquad \forall\, i, \tag{23}$$

where $q(x_i|s_i)$ and $q(s_i)$ are defined as follows:

$$q(x_i|s_i) = \mathcal{N}(m(s_i), \Gamma(s_i)), \tag{24}$$

$$q(s_i) \propto \frac{1}{\sqrt{\sigma_n^2 + \sigma^2(s_i)}} \exp\left\{ \frac{\sigma^2(s_i)}{2\sigma_n^2} \frac{\mathbf{r}_i^T \mathbf{d}_i \mathbf{d}_i^T \mathbf{r}_i}{\sigma_n^2 + \sigma^2(s_i)} \right\} p(s_i), \tag{25}$$

and

$$\Gamma(s_i) = \frac{\sigma^2(s_i)\sigma_n^2}{\sigma^2(s_i) + \sigma_n^2}, \tag{26}$$

$$m(s_i) = \frac{\sigma^2(s_i)}{\sigma_n^2 + \sigma^2(s_i)} \mathbf{d}_i^T\, \mathbf{r}_i, \tag{27}$$

$$\mathbf{r}_i = \mathbf{y} - \sum_{j \neq i} \langle x_j \rangle_{q(x_j, s_j)} \mathbf{d}_j. \tag{28}$$

Using (22), an approximation of $p(\mathbf{x}|\mathbf{y})$ can be computed as follows

$$p(\mathbf{x}|\mathbf{y}) \simeq \prod_i \sum_{s_i} q(x_i, s_i), \tag{29}$$

and the solution of (8) can then be approximated as

$$\hat{x}_i \simeq \langle x_i \rangle_{q(x_i)} = \sum_{s_i} m(s_i) q(s_i) \qquad \forall\, i. \tag{30}$$

In the sequel, we will refer to the procedure defined by (22)-(27) as VBSR2. This algorithm can actually be regarded as a soft version of MP: at each iteration, the estimate of the sparse vector $\mathbf{x}$ is updated by a weighted version of the projection of the current residual, see (27)-(30). Similarly to MP, the order in which probabilities $q(x_i, s_i)$ are updated plays therefore an important role in the performance of the algorithm. At each iteration, we choose to update the factor with the highest probability of activity $q(s_i)$.

The computational complexity of VBSR2 is dominated by the evaluation of (27) and is therefore of order M per iteration. This complexity is similar to the complexity of MP or GP.

3.4. Combination of mean-field approximations

The MF approximation implies to break the statistical dependencies between some of the variables. For example, the independence between $\mathbf{x}$ and $\mathbf{s}$ (resp. all the couples (x_i, s_i)) is forced in (17) (resp. (22)). Although these independence assumptions simplify the computation of $q(\mathbf{x}, \mathbf{s})$, they also lead to the loss of some statistical information. In this section we propose a heuristic algorithm which intends to reduce this loss of information by taking benefit (up to a point) from both decompositions (17) and (22).

The algorithm is defined as a combination of VBSR1 and VBSR2 updates:

1. $q^{(n)}(\mathbf{x}) = \mathcal{N}(m, \Gamma)$ where m and Γ are defined in (20)-(21).
2. $q^{(n)}(\mathbf{s}) = \prod_i q(s_i)$ where $q(s_i)$ are computed from (25)-(27) by using $\langle x_j \rangle_{q(x_j, s_j)} = \langle x_j \rangle_{q^{(n)}(\mathbf{x})}$ in (27).

The first step thus relies on assumption (17), the second one on assumption (22). This algorithm will be referred to as VBSR3 in the sequel. The complexity of VBSR3 is dominated by the matrix inversion (20) and scales therefore with N^3. This complexity is similar to the one of VBSR1. It is important to mention that the VBSR3 update equations do not define a VB-EM algorithm and the convergence of VBSR3 is therefore not theoretically ensured. However, we will see in the next section that VBSR3 leads to good empirical results.

4. SIMULATIONS

In this section, we study the performance of the proposed SR algorithms by extensive computer simulations. We follow the same methodology as in [5] to assess the performance of the SR algorithms: we calculate the empirical frequency of correct reconstruction versus the number of non-zero coefficients in $\mathbf{x}$, say K. We assume that a vector has been correctly reconstructed when the amplitude of the error reconstruction on each non-zero coefficient is lower than 10^{-4}.

Fig. 1 illustrates the performance achieved by VBSR1, VBSR2 and VBSR3. The performance of other standard SR algorithms (MP, OMP, BP, SP and RVM) are also reported for the sake of comparison. We use the following parameters for the generation of these curves: $N = 128$, $M = 256$, $\sigma_n^2 = 10^{-5}$. The elements of the dictionary are *i.i.d* realizations of a zero-mean Gaussian distribution with variance N^{-1}. The positions of the non-zero coefficients are drawn uniformly at random. The amplitude of the active (resp. inactive) coefficients are generated from a zero-mean Gaussian with variance $\sigma^2(s_i = 1) = 10$ (resp. $\sigma^2(s_i = 0) = 10^{-8}$). For each point of simulation, we run 200 trials.

[4] In [16], Attias applied the mean-field approximation (22) to a BG model in the context of blind source separation. However, no assumptions about the sparsity of the sources were made in this paper.

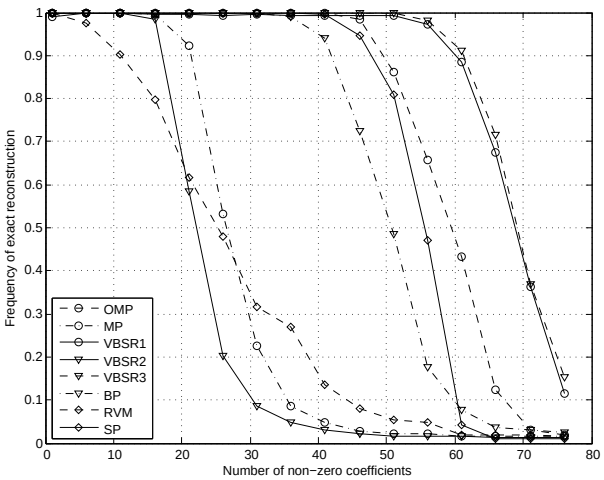

Fig. 1. Frequency of exact reconstruction versus number of nonzero coefficients; $N = 128$, $M = 256$, $\sigma_n^2 = 10^{-5}$, $\sigma^2(s_i = 0) = 10^{-8}$, $\sigma^2(s_i = 1) = 10\ \forall i$.

MP and OMP are run until the l_2-norm of the residual drops below $\sqrt{N\sigma_n^2}$. The probabilities of activity used by the VBSR algorithms are set to $p(s_i = 1) = K/M\ \forall i$. We noticed that the performance of VBSR1 and VBSR3 can be greatly improved by progressively decreasing the variance on the inactive coefficients. We used the following strategy:

$$(\sigma^2(s_i = 0))^{(n)} = 0.8\,\sigma^2(s_i = 1)\alpha^n + \sigma^2(s_i = 0) \qquad \forall i, \quad (31)$$

where n is the iteration number and $\alpha < 1$.

A good figure of merit of SR algorithms is their *critical sparsity*, *i.e.*, the maximum number of nonzero coefficients for which the original sparse vector $\mathbf{x}$ can be reconstructed with frequency one. As far as our simulation setup is concerned, we see from Fig. 1 that both VBSR1 and VBSR3 clearly outperform the other SR algorithms: VBSR1 and VBSR3 start failing for $K \geq 55$ whereas SP (resp. BP) has its critical sparsity located around $K = 45$ (resp. $K = 35$). Note that, if $\mathbf{D}$ is a rank-N matrix, it is well-known that the optimal (but intractable) estimator which computes the exact solution of (1) can recover any sparse vector if $K \leq N/2 = 64$. VBSR1 and VBSR3 are getting very close to this limit since they can recover roughly 70% of the active coefficients when $K = 64$.

On the other hand, the performance of VBSR2 is quite poor and similar to the one of MP. This is due to the fact that (22) is a poor approximation of the true a posteriori probability $p(\mathbf{x}|\mathbf{y})$.

5. CONCLUSION

In this paper, we consider the sparse representation problem within a Bernoulli-Gaussian Bayesian framework. We propose several tractable solutions to the Bayesian inference problem by resorting to mean-field variational approximations and the VB-EM algorithm. The resulting SR algorithms are shown to have very good performance over a wide range of sparsity levels. In particular, they significantly improve the critical sparsity upon state-of-the-art SR algorithms. The complexity of our best algorithm evolves as N^3, which may be too large for some large-scale applications. However, strong connections are made between the proposed algorithms and low-complexity SR algorithms such as CoSaMP/SP. This observation paves the way for the design of low-complexity versions of VBSR algorithms and is part of ongoing work.

6. REFERENCES

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

[2] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Ann. Asilomar Conf. Signals, Systems, and Computers*, 1993.

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

[4] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," available at arXiv:0803.2393v2, April 2008.

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

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

[7] I. Gorodnitsky and D. R. Bhaskar, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, March 1997.

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

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

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

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

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

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

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

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

[16] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 44, pp. 803–851, 1999.