Exploiting Statistical Dependencies in Sparse Representations for Signal Recovery

Tomer Faktor, Yonina C. Eldar, *Senior Member, IEEE*, and Michael Elad, *Senior Member, IEEE*

***Abstract*—Signal modeling lies at the core of numerous signal and image processing applications. A recent approach that has drawn considerable attention is sparse representation modeling, in which the signal is assumed to be generated as a combination of a few atoms from a given dictionary. In this work we consider a Bayesian setting and go beyond the classic assumption of independence between the atoms. The main goal of this paper is to introduce a statistical model that takes such dependencies into account and show how this model can be used for sparse signal recovery. We follow the suggestion of two recent works and assume that the sparsity pattern is modeled by a Boltzmann machine, a commonly used graphical model. We show that for general dependency models, exact MAP estimation of the sparse representation becomes computationally complex. To simplify the computations, we propose a greedy approximation for the MAP estimator. We then consider a special case where exact MAP is feasible, by assuming that the dictionary is unitary and the dependency model corresponds to a certain sparse graph. Exploiting this structure, we develop an efficient message-passing algorithm that recovers the underlying signal. The effectiveness of our developed pursuit methods is demonstrated on synthetic signals, where we compare the denoising performance to that of previous recovery methods that do not exploit the statistical dependencies. Finally, when the model parameters defining the underlying graph are unknown, we suggest an algorithm that learns these parameters directly from the data, leading to an iterative scheme for adaptive sparse signal recovery.**

***Index Terms*—Sparse representations, signal synthesis, Bayesian estimation, MAP, MRF, Boltzmann machine, greedy pursuit, unitary dictionary, decomposable model, message passing, pseudo-likelihood, SESOP.**

I. Introduction

Signal modeling based on sparse representations is used in numerous signal and image processing applications, such as denoising, restoration, source separation, compression and sampling (for a comprehensive review see [1]). The basic ingredients of a typical generation model are a sparse representation x, a dictionary that is applied on this representation and additive noise. Each of these ingredients can be regarded deterministic or random, leading to different recovery methods and performance guarantees. In this paper we focus on the modeling of the sparse representation. The classical approach to sparse recovery considers a deterministic sparse representation and signal recovery is formulated as a deterministic optimization problem. Some examples include greedy pursuit algorithms like orthogonal matching pursuit (OMP) and CoSaMP, and convex relaxations like basis pursuit denoising and the Dantzig selector (for comprehensive reviews see [1], [2]).

Recent works [3], [4], [5], [6], [7], [8] suggested imposing additional assumptions on the support of x (the sparsity pattern), which is still regarded deterministic there. These works show that using structured sparsity models that go beyond simple sparsity can boost the performance of standard sparse recovery algorithms in many cases. Two typical examples for such models are wavelet trees [3] and block-sparsity [5], [6]. The first accounts for the fact that the large wavelet coefficients of piecewise smooth signals and images tend to live on a rooted, connected tree structure [9]. The second model is based on the assumption that the signal exhibits special structure in the form of the nonzero coefficients occurring in clusters. This is a special case of a more general model, where the signal is assumed to lie in a union of subspaces [4], [5]. Block-sparsity arises naturally in many setups, such as recovery of multi-band signals [10], [11] and the multiple measurement vector (MMV) problem. However, there are many other setups in which sparse elements do not fit such simple models.

In many applications it can be difficult to provide one deterministic model that describes all signals of interest. For example, in the special case of wavelet trees it is well known that statistical models, such as hidden Markov trees (HMTs) [12], is more reliable than a deterministic one. Guided by the observation that statistical models can often be more powerful than deterministic ones, it is natural to consider more general Bayesian modeling, in which the sparse representation is assumed to be a random vector. Many sparsity-favoring priors for the representation coefficients have been suggested in statistics, such as the Laplace prior, "spike-and-slab" (mixture of narrow and wide Gaussian distributions) and Student's t distribution (for a comprehensive review see [13]). However, the representation coefficients are typically assumed to be independent of each other.

Here we are interested in Bayesian modeling that takes into account not only the values of the representation coefficients, but also their sparsity pattern (the support of x). In this framework sparsity is achieved by placing a prior distribution on the support, and the representation coefficients are modeled through a conditional distribution given the support. The most simple prior for the support assumes that the entries of the sparsity pattern are independent and identically distributed (i.i.d.) (see e.g. [14]). However, in practice, atoms in the dictionary are often not used with the same frequency. To account for this behavior, we can relax the assumption that

T. Faktor and Y. C. Eldar are with the Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel (e-mail: {tomerfa@tx,yonina@ee}.technion.ac.il). M. Elad is with the Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel (e-mail: elad@cs.technion.ac.il).

This work was supported in part by the Israel Science Foundation under Grants 1081/07 and 599/08, and by the European Commission's FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (grant agreement no. 216715) and FP7-FET program, SMALL project (grant agreement no. 225913).

the entries are identically distributed and assign different probabilities to be turned "on" for each entry (see e.g. [15]).

Besides the modeling aspect, another key ingredient in Bayesian formulations is the design objective. Two popular techniques are maximum *a posteriori* (MAP) and minimum mean square error (MMSE) estimators. Typically these estimators are computationally complex, so that they can only be approximated. For example, approximate MAP estimation can be performed using a wide range of inference methods, such as the relevance vector machine [16] and Markov chain Monte Carlo (MCMC) [17]. Such estimators are derived in [13], [18] based on sparsity-favoring priors on x and approximate inference methods. In [14], [19] approximate MMSE estimators are developed, based on an i.i.d prior on the support. Finally, in the special case of a square and unitary dictionary, assuming independent entries in the support and Gaussian coefficients, it is well known that the exact MAP and MMSE estimators can be easily computed [15].

Independence between the entries in the support can be a useful assumption, as it keeps the computational complexity low and the performance analysis simple. Nevertheless, this assumption can be quite restrictive and leads to loss of representation power. Real-life signals exhibit significant connections between the atoms in the dictionary used for their synthesis. For example, it is well known that when image patches are represented using the discrete cosine transform (DCT) or a wavelet transform, the locations of the large coefficients are strongly correlated. Several recent works [7], [20], [21], [22], [23] have made attempts to go beyond the classic assumption of independence and suggested statistical models that take dependencies into account. The special case of wavelet trees has been addressed in [7], [20], where HMTs are merged into standard sparse recovery algorithms, in order to improve some of their stages and lead to more reliable recovery. Another statistical model designed to capture the tree structure for wavelet coefficients, was suggested in [21]. An approximate MAP estimator was developed there based on this model and MCMC inference.

Here we consider more general dependency models based on undirected graphs, which are also referred as Markov random fields (MRFs). Graphical models [24] provide a full and concise description for the prior distribution on the support and allow to perform probabilistic inference using powerful methods developed in the field of graph theory. Two examples of such techniques that are widely used for MAP estimation are belief propagation [24] and graph cuts [25]. In [22] the authors propose a generative model for sparse representations that is based on a Boltzmann Machine (BM), an appealing MRF for the prior on the support. This allows for introducing the concept of interactions in a general sparse coding model. An approximate MAP estimator is then developed by means of Gibbs sampling and simulated annealing [17]. Note however that these are general-purpose optimization techniques, which often suffer from high computational effort and a slow convergence rate. In [23] a BM prior on the support is used in order to improve the CoSaMP algorithm. Note however that the Bayesian model is used only in one stage of the algorithm (the one that obtains an estimate for the support given x), whereas all the other stages, including the stopping rule for the algorithm, remain unchanged.

The current paper is aimed at further exploring the BM-based model proposed in [22]. Our main contributions include exploring settings where exact MAP estimation is computationally feasible and designing specialized methods for both MAP estimation and model estimation. We develop an efficient message-passing algorithm for signal recovery which obtains the exact MAP estimate under some additional modeling assumptions. We also suggest a greedy algorithm for signal recovery which approximates the MAP estimator for the general BM-based model. The proposed greedy algorithm takes into account the statistical generative model throughout all its stages, including the stopping rule. The main contributions and drawbacks of the two recent works which used the BM-based model [22], [23], as well as the differences between these works and the current work, will be discussed in more detail in Section IX.

The paper is organized as follows. In Section II we motivate the need for inserting probabilistic dependencies between elements in the support by considering sparse representations of image patches over a DCT dictionary. In Section III we introduce useful notions and tools from the graphical models field and explore the BM prior. Section IV defines the signal model and the MAP estimation problem. In Section V we develop a greedy approximation of the MAP estimator for the BM prior. We then present setups where the problem can be solved exactly and develop an efficient algorithm for obtaining the exact solution in Section VI. We explore the performance of these two algorithms through synthetic experiments in Section VII. Estimation of the model parameters and adaptive sparse signal recovery are addressed in Section VIII. Finally, we discuss relations to past works in Section IX.

II. Motivation

In this section we provide motivation for inserting probabilistic dependencies between elements in the support. We consider a set of $N = 100,000$ patches of size 8-by-8 that are extracted out of several noise-free natural images. For each patch, we perform a preliminary stage of DC removal by subtracting the average value of the patch, and then obtain sparse representations of these patches over an overcomplete DCT dictionary of size 64-by-256 (n-by-m) using the OMP algorithm. We consider a model error of $\sigma = 2$, so that OMP stops when the residual error falls below $\epsilon = \sqrt{n}\sigma = 16$. We then compute the empirical marginal distributions for each of the dictionary atoms and for all pairs of atoms, namely we approximate $\Pr(S_i = 1),\ i = 1,\ldots,m$ and $\Pr(S_i = 1, S_j = 1),\ i = 1,\ldots,m-1,\ j > i$, where S is a binary vector of size m and $S_i = 1$ denotes that the ith atom is being used. The empirical conditional probability $\Pr(S_i = 1|S_j = 1)$ can then be computed as the ratio between $\Pr(S_i = 1, S_j = 1)$ and $\Pr(S_j = 1)$.

We address several assumptions that are commonly used in the sparse recovery field and suggest validity tests for each of them. The first assumption is that the elements in the support vector are identically distributed, namely $\Pr(S_i =$

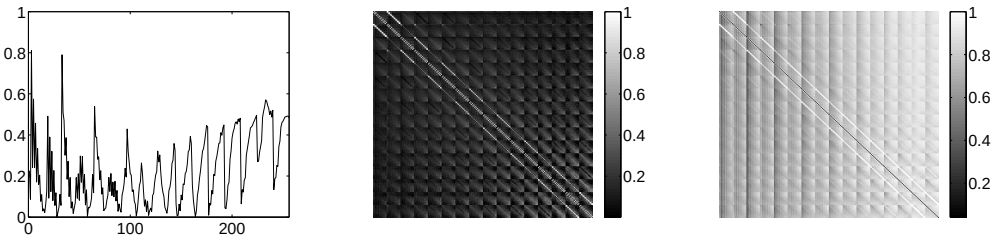


Figure 1. Validity tests for several assumptions on the support vector: identical distributions, independency and block-sparsity. Left: A plot of R, Middle: An image of U, Right: An image of V.

$1) = p$ for all i, where $0 \leq p \leq 1$ is some constant. This assumption can be examined by comparing the marginal probabilities $\Pr(S_i = 1)$ for each atom. The second assumption is independency between elements in the support. The independency assumption between atoms i and j implies that $\Pr(S_i = 1|S_j = 1) = \Pr(S_i = 1)$. Therefore, we can test for independency by comparing the marginal and conditional probabilities for each pair of atoms. Next we turn to the block-sparsity assumption. Assuming that i and j are in the same cluster implies that the conditional probabilities $\Pr(S_i = 1|S_j = 1)$ and $\Pr(S_j = 1|S_i = 1)$ are near 1.

To examine the validity of each of the above-mentioned assumptions, we compute the variables

$$\begin{aligned} R_i &= \left|\log_{10}\left(\frac{\Pr(S_i=1)}{p}\right)\right|,\ 1 \leq i \leq m \\ U_{i,j} &= \left|\log_{10}\left(\frac{\Pr(S_i=1|S_j=1)}{\Pr(S_i=1)} + \delta\right)\right|,\ 1 \leq i,j \leq m \\ V_{i,j} &= \left|\log_{10}\left(\Pr(S_i=1|S_j=1) + \delta\right)\right|,\ 1 \leq i,j \leq m \end{aligned} \quad (1)$$

where p denotes the average probability of an entry to be turned "on" , namely $p \triangleq \frac{1}{m}\sum_{l=1}^{m}\Pr(S_l = 1)$, R is a vector of size m and U, V are matrices of size m-by-m. We use $\delta = 0.1$, so that for $\Pr(S_i = 1|S_j = 1) = 0$ we get a value 1 in $U_{i,j}$ and $V_{i,j}$ (i and j denote the row and column indexes respectively). In each of the functions in (1) a near-zero result implies that the corresponding assumption is valid; as we go further away from zero the validity of the assumption decreases.

The results are shown in Fig. 1. On the left we plot the values in R. This plot demonstrates that the individual frequencies can be very far from the average one. Consequently, the DCT atoms are used with varying frequencies. The matrix U is displayed in the middle. The black color, which corresponds to near-zero values, is dominant. This illustrates that the independency assumption is satisfactory for many pairs of DCT atoms. However, some pairs exhibit significant interactions (see the white diagonals near the main diagonal and the bright spots). The image on the right displays the matrix V, which is dominated by the white color, corresponding to near-one values. High values in the entries $V_{i,j}$ or $V_{j,i}$ indicate that it is not reasonable to assume that the corresponding atoms belong to the same cluster in a block-sparse model (regardless of the block sizes). Since this is the case for most pairs of DCT atoms, we conclude the block-sparsity approach does not capture the dependencies well in this example.

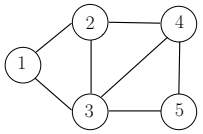
It is interesting to note that while the OMP algorithm reveals different frequencies of appearance for the atoms and significant correlations between pairs of atoms, it in fact makes no use of these properties. Therefore, it seems plausible that a stochastic model that will capture the different nature of each atom, as well as the important interactions between the atoms, can lead to improved performance. In this paper we will show how this can be accomplished in a flexible and adaptive manner.

III. Graphical Models

The main goal of this paper is using graphical models for representing statistical dependencies between elements in the sparsity pattern and developing efficient sparse recovery algorithms based on this modeling. In order to set the ground for the signal model and the recovery algorithms, we provide some necessary notions and methods from the vast literature on graphical models. We begin by presenting MRFs and explain how they can be used for describing statistical dependencies. We then focus on the BM, a widely used MRF, explore its properties and explain how it can serve as a useful and powerful prior on the sparsity pattern. For computational purposes we may want to relax the dependency model. One possible relaxation, which often reduces computational complexity and still bares considerable representation power, is decomposable models. Finally, we present a powerful method for probabilistic inference in decomposable models, coined belief propagation. Decomposability will be a modeling assumption in Section VI and the algorithm we propose in Section VI-B will be based on belief propagation techniques.

A. Representing Statistical Dependencies by MRFs

In this subsection we briefly review MRFs and how they can used to represent statistical dependencies. This review is mainly based on [24]. A graphical model is defined by its structural and parametric components. The structural component is the graph $G = (V, \varepsilon)$ where V is a set of nodes (vertices) and ε is a set of undirected edges (links between the nodes). In a graphical model there is a one-to-one mapping between nodes $\{1, 2, \ldots, m\}$ and random variables $\{S_1, S_2, \ldots, S_m\}$. Let S_A, S_B, S_C stand for three disjoint subsets of nodes. We say that S_A is independent of S_C given S_B if S_B separates S_A from S_C, namely all paths between a node in S_A and a node in S_C pass via a node in S_B. Thus, simple graph separation is equivalent to conditional independence. The structure can be used to obtain all the global conditional independence relations of the probabilistic model. By "global" we mean that conditional independence



(a) Graph

$$\begin{pmatrix} 0 & W_{12} & W_{13} & 0 & 0 \\ W_{12} & 0 & W_{23} & W_{24} & 0 \\ W_{13} & W_{23} & 0 & W_{34} & W_{35} \\ 0 & W_{24} & W_{34} & 0 & W_{45} \\ 0 & 0 & W_{35} & W_{45} & 0 \end{pmatrix}$$

(b) Interaction matrix

Figure 2. A simple dependency model for 5 variables. This is a chordal graph with 3 missing edges. The interaction matrix in the corresponding BM is banded.

holds for all variable assignments and does not depend on numerical specifications. For a visual demonstration see Fig. 2(a); using the above definition it easy to verify for example that S_1 is independent of S_4, S_5 given S_2, S_3.

Turning to the parametric component, note that the joint probability distribution is represented by a local parametrization. More specifically, we use a product of local nonnegative compatibility functions, which are referred to as potentials. The essence of locality becomes clearer if we define the notion of cliques. A clique is defined as a fully-connected subset of nodes in the graph. If S_i and S_j are linked, they appear together in a clique and thus we can achieve dependence between them by defining a potential function on that clique. The maximal cliques of a graph are the cliques that cannot be extended to include additional nodes without losing the property of being fully connected. Since all cliques are subsets of one or more maximal cliques, we can restrict ourselves to maximal cliques without loss of generality. For example, in Fig. 2(a) the maximal cliques are $C_1 = \{1, 2, 3\}$, $C_2 = \{2, 3, 4\}$ and $C_3 = \{3, 4, 5\}$. To each maximal clique C we assign a nonnegative potential $\Psi_C(S_C)$. The joint probability is then given as a product of these potentials, up to a normalization factor Z:

$$\Pr(S) \triangleq \frac{1}{Z} \prod_C \Psi_C(S_C). \tag{2}$$

If the potentials are taken from the exponential family, namely $\Psi_C(S_C) = \exp\{-E_C(S_C)\}$, then $\Pr(S) = \frac{1}{Z}\exp\{-E(S)\}$, where $E(S) = \sum_C E_C(S_C)$ is the energy of the system.

B. The Boltzmann Machine

In this subsection we focus on the BM, a widely used MRF. We are about to show that this can serve as a useful and powerful prior on the sparsity pattern. The BM distribution is given by:

$$\Pr(S) = \frac{1}{Z}\exp\left(b^T S + \frac{1}{2} S^T W S\right), \tag{3}$$

where S is a binary vector of size m with values in $\{-1, 1\}^m$, W is symmetric and Z is a partition function of the Boltzmann parameters W, b that normalizes the distribution. We can further assume that the entries on the main diagonal of W are zero, since they contribute a constant to the function $S^T W S$. In this work the BM will be used as a prior on the support of a sparse representation: $S_i = 1$ implies that the ith atom is used for the representation, whereas for $S_i = -1$ this atom is not used.

The BM is a special case of the exponential family with an energy function $E(S) = -b^T S - \frac{1}{2} S^T W S$. The BM distribution can be easily represented by a MRF - a bias b_i is associated with a node i and a nonzero entry W_{ij} in the interaction matrix results in an edge connecting nodes i and j with the specified weight. Consequently, the zero entries in W have the simple interpretation of missing edges in the corresponding undirected graph. This means that the sparsity pattern of W is directly linked to the sparsity of the graph structure. From graph separation we get that if $W_{ij} = 0$ then S_i and S_j are statistically independent given all their neighbors $\{S_l\}_{l \in\ N(i) \cup N(j),\ l \neq i,j}$. For example, if the matrix W corresponds to the undirected graph that appears in Fig. 2(a) then $W_{14} = W_{15} = W_{25} = 0$. This matrix is shown in Fig. 2(b).

The maximal cliques in the BM are denoted by $C_1, \ldots, C_P$ and we would like to assign potential functions $\{\Psi_{C_i}(S_{C_i})\}_{i=1}^P$ to these cliques that will satisfy the requirement $\exp\left(b^T S + \frac{1}{2} S^T W S\right) = \prod_{i=1}^P \Psi_{C_i}(S_{C_i})$. One possible choice is to assign each of the terms in $E(S)$ using a prespecified order of the cliques: $b_i S_i$ is assigned to the clique that consists of S_i and appears last in the order and a non-zero term $W_{ij} S_i S_j$ is assigned to the clique that consists of S_i, S_j and appears last in the order.

Next, we turn to explore the intuitive meaning of the Boltzmann parameters. In the simple case of $W = 0$, the BM distribution becomes $\Pr(S) = \frac{1}{Z}\prod_{i=1}^m \exp(b_i S_i)$. Consequently, $\{S_i\}_{i=1}^m$ are statistically independent and this assumption is referred as "independency". Using straight forward computations we get $\Pr(S_i = -1) = \exp(-2b_i)\Pr(S_i = 1)$ for $i = 1, \ldots, m$. Since $\Pr(S_i = -1) + \Pr(S_i = 1) = 1$, S_i has the following marginal probability to be turned "on":

$$p_i \triangleq \Pr(S_i = 1) = \frac{1}{1 + \exp(-2b_i)},\ 1 \leq i \leq m. \tag{4}$$

When W is nonzero, (4) no longer holds. However, the simple intuition that S_i tends to be turned "off" as b_i becomes more negative, remains true.

We would now like to understand how to describe correlations between elements in S. To this end we focus on the simple case of a matrix W of size 2-by-2, consisting of one parameter W_{12}, and provide an exact analysis for this setup. In order to simplify notations, from now on we use $p_{i|j}(u|v)$ to denote $\Pr(S_i = u | S_j = v)$. Using these notations we can write down the following relation for the simple case of a pair of nodes:

$$p_1 = p_{1|2}(1|1)p_2 + p_{1|2}(1|-1)(1 - p_2), \tag{5}$$

where

$$\begin{aligned} p_{1|2}(1|1) &= \frac{1}{1 + \exp(-2b_1 - 2W_{12})} \\ p_{1|2}(1|-1) &= \frac{1}{1 + \exp(-2b_1 + 2W_{12})}. \end{aligned} \tag{6}$$

From (5) we see that p_1 is a convex combination of $p_{1|2}(1|-1)$ and $p_{1|2}(1|1)$. Hence, for $W_{12} > 0$ we have $p_{1|2}(1|-1) < p_1 < p_{1|2}(1|1)$ and for $W_{12} < 0$ we have $p_{1|2}(1|1) < p_1 < p_{1|2}(1|-1)$.

For a general matrix W these relations are no longer strictly accurate. However, they serve as useful rules of thumb: for

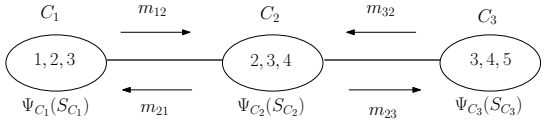


Figure 3. A clique tree which is constructed for the graph that appears in Fig. 2. In this case the clique tree takes the form of a simple chain of size 3. Potential functions are defined for each of the cliques and exact probabilistic inference is performed by message-passing.

an "excitatory" interaction ($W_{ij} > 0$) S_i and S_j tend to be turned "on" ("off") together, and for an "inhibitory" interaction ($W_{ij} < 0$) S_i and S_j tend to be in opposite states. The intuition into the Boltzmann parameters provides some guidelines as to how the BM prior can be used for sparse representations. If the values of the biases in the vector b are negative "enough" and there are few strong excitatory interactions, then the mean cardinality of the support tends to be small. This reveals some of the power of the BM as a prior on the support in the signal model. It can achieve sparsity and at the same time capture statistical dependencies and independencies in the sparsity pattern.

To conclude this section, note that standard sparsity-favoring models can be obtained as special cases of the BM model. For $W = 0$ and $b_i = \frac{1}{2}\ln\left(\frac{p}{1-p}\right)$ for all i, which correspond to an i.i.d. prior, the cardinality k has a Binomial distribution, namely $k \sim Bin(p, m)$. For a low value of p the cardinalities are typically much smaller than m, so that plain sparsity is achieved. BM can also describe a block-sparsity structure: Assuming that the first k_1 entries in S correspond to the first block, the next k_2 to the second block, etc., the interaction matrix W should be block-diagonal with "large" and positive entries within each block. The entries in b should be chosen as mentioned above to encourage sparsity.

C. Decomposable Graphical Models

We now consider decomposability in graphical models [24], [26]. A triplet $\{A, B, C\}$ of disjoint subsets of nodes is a decomposition of a graph if its union covers all the set V, B separates A from C and B is fully-connected. It follows that a graphical model is regarded as decomposable if it can be recursively decomposed into its maximal cliques, where the separators are the intersections between the cliques. It is well known that a decomposable graph is necessarily chordal, i.e. every cycle of length four or more in the graph has a an edge joining two nonconsecutive nodes. Consequently, for a given MRF we can apply a simple graphical test to verify that it is decomposable.

In Section VI we consider decomposable BMs. This assumption implies that the matrix W corresponds to a chordal graph. We now provide some important examples for decomposable graphical models and their corresponding interaction matrices. Note that a graph which contains no cycles of length four is obviously chordal as it satisfies the required property in a trivial sense. It follows that a graph with no edges, a graph consisting of non-overlapping cliques and a tree are all chordal. The first example is the most trivial chordal graph and corresponds to $W = 0$. The second corresponds to a block-diagonal matrix and as we explained in Section III-B it can describe a block-sparsity structure. Tree structures are widely used in applications that are based on a multiscale framework. A visual demonstration of the corresponding matrix is shown in [26].

Another common decomposable model corresponds to a banded interaction matrix. In an Lth order banded matrix only the $2L + 1$ principle diagonals consist of nonzero elements. Assuming that the main diagonal is set to zero, we have that there can be at most $(2m-(L+1))L$ nonzero entries in an Lth order banded W, instead of $m^2 - m$ nonzeros as in a general interaction matrix. Consequently, the sparsity ratio of W is of order L/m. This matrix corresponds to a chordal graph with cliques $C_i = \{S_i, \ldots, S_{i+L}\}$, $i = 1, \ldots, m-L$. For example, the matrix in Fig. 2(b) is a second order banded matrix of size 5-by-5. This matrix corresponds to a chordal graph (see Fig. 2(a)) with three cliques.

Chordal graphs serve as a natural extension to trees. It is well known [24] that the cliques of a chordal graph can be arranged in a clique tree, which is called a junction tree. In a junction tree T each clique serves as a vertex and any two cliques containing a node v are either adjacent in T or connected by a path made entirely of cliques containing v. For a visual demonstration see Fig. 3, where a clique tree is constructed for the chordal graph of Fig. 2(a). In this case where the interaction matrix is banded, the clique tree is simply a chain. It can easily be verified that this is in fact true for a banded interaction matrix of any order.

We now turn to describe belief propagation, a powerful method for probabilistic inference tasks like computation of single node marginal distributions and finding the most probable configuration. Exact probabilistic inference can become computationally infeasible for general dependency models as it requires a summation or maximization over all possible configurations of the variables. For example, in a general graphical model with m binary variables the complexity of exact inference grows exponentially with m. However, for when the graph structure is sparse, one can often exploit the sparsity in order to reduce this complexity. The inference tasks mentioned above can often be performed efficiently using belief propagation techniques [24]. More specifically, in a decomposable MRF exact inference takes the form of a message-passing algorithm, where intermediate factors are sent as messages along the edges of the junction tree (see for example the messages passed along the chain in Fig. 3). For more details on message passing see [24].

The complexity of exact inference via message-passing strongly depends on the tree-width of the graph. In a decomposable model this is defined as the size of the largest maximal clique minus one. For example, in the special case of a BM with an Lth order banded W we have that the tree-width is L. We can conclude that for a decomposable model there is an obvious tradeoff between computational complexity and representation power. For example, in the special case of an Lth order interaction matrix the computational complexity of exact inference decreases with L, but at the same time the graphical model captures fewer interactions. Nevertheless, decomposable models can serve as a useful relaxation for a general dependency model, as they can achieve a substantial

decrease in the complexity of exact inference, while still capturing the significant interactions.

IV. BM Generative Model

In this section we use the BM for constructing a stochastic generative signal model. We consider a signal y which is modeled as $y = Ax + e$, where A is the dictionary of size n-by-m, x is a sparse representation over this dictionary and e is additive white Gaussian noise (AWGN) with variance σ_e^2. This is a very common and long-studied model in signal and image processing. Various works that are based on this model differ in their modeling for the sparse representation x. We denote the sparsity pattern by $S \in \{-1, 1\}^m$, where $S_i = 1$ implies that the index i belongs to the support of x, whereas $S_i = -1$ implies that $x_i = 0$. The nonzero coefficients of x are denoted by x_s, where s is the support of x. Following [22] we consider a BM prior for S and a Gaussian distribution with zero mean and variance $\sigma_{x,i}^2$ for each nonzero representation coefficient x_i. Note that the variances of the non-zero representation coefficients are atom-dependent. It follows that the conditional distribution of x_s given the support s is

$$\Pr(x_s|s) = \frac{1}{\det\left(2\pi\Sigma_s\right)^{1/2}} \exp\left\{-\frac{1}{2} x_s^T \Sigma_s^{-1} x_s\right\} \quad (7)$$

where Σ_s is a $k \times k$ diagonal matrix with diagonal elements $(\Sigma_s)_{i,i} = \sigma_{x,s_i}^2$, where k is the cardinality of the support s. Using the assumption that the noise is AWGN we can also write down the conditional distribution for the signal y given its sparse representation:

$$\Pr(y|x_s, s) = \frac{1}{(2\pi\sigma_e^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma_e^2}\|y - A_s x_s\|_2^2\right\}. \quad (8)$$

Our goal is to recover x given y. Note however that $\Pr(x|y)$ is a mixture of a discrete distribution for $x = 0$ and a continuous distribution for all nonzero values of x. Consequently, given y we have that $x = 0$ with a nonzero probability, whereas for any nonzero vector v the event $x = v$ occurs with probability zero. It follows that the MAP estimator for x given y leads to the trivial solution $x = 0$, rendering it useless. The distribution $\Pr(s|y)$ however is a discrete one. Therefore, we suggest to first perform MAP estimation of s given y and then proceed with MAP estimation of x given y and the estimated support $\hat{s}$ [15].

We begin by developing an expression for $\Pr(y|s)$ by integrating over all possible values of $x_s \in \mathbb{R}^k$:

$$\begin{aligned}\Pr(y|s) &= \int_{x_s \in \mathbb{R}^k} \Pr(y|x_s, s) \Pr(x_s|s) dx_s \quad (9)\\ &= C \frac{1}{\det\left(\frac{1}{\sigma_e^2} A_s^T A_s \Sigma_s + I\right)^{1/2}} \exp\left\{\frac{1}{2\sigma_e^2} y^T A_s Q_s^{-1} A_s^T y\right\}\end{aligned}$$

where $C = {}^1/_{(2\pi\sigma_e^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma_e^2}\|y\|_2^2\right\}$ is a constant and $Q_s = A_s^T A_s + \sigma_e^2 \Sigma_s^{-1}$. This leads to the following estimator for the support:

$$\begin{aligned}\hat{s}_{MAP} =& \underset{s\in\Omega}{\text{argmax}} \Pr(s|y) = \underset{s\in\Omega}{\text{argmax}} \Pr(y|s) \Pr(s)\\ =& \underset{s\in\Omega}{\text{argmax}} \frac{1}{2\sigma_e^2} y^T A_s Q_s^{-1} A_s^T y - \quad (10)\\ & \frac{1}{2}\ln\left(\det\left(Q_s\right)\right) + \frac{1}{2} S^T W S + \left(b - \frac{1}{4} v\right)^T S\end{aligned}$$

where $v_i = \ln\left(\sigma_{x,i}^2/\sigma_e^2\right)$ and S depends on s through $S_i = 2\cdot\mathbf{1}[i \in s] - 1$ for all i, with $\mathbf{1}[\cdot]$ denoting the indicator function. The feasible set Ω denotes all 2^m possible supports. In terms of S, this is the set of all vectors satisfying $S_i^2 = 1$ for all i. Note that for an empty support the two first terms in (10) vanish.

Once we have an estimate $\hat{s} = \hat{s}_{MAP}$ of the support, we can compute a MAP estimator of x using the same formula as in the oracle estimator (see [15]):

$$\hat{x}_{s_{MAP}} = \underset{x_s \in \mathbb{R}^k}{\text{argmax}} \Pr(x|y, \hat{s}) = \left(A_{\hat{s}}^T A_{\hat{s}} + \sigma_e^2 \Sigma_{\hat{s}}^{-1}\right)^{-1} A_{\hat{s}}^T y. \quad (11)$$

In the sequel we first focus on the case where all model parameters - the Boltzmann parameters W, b, the variances $\left\{\sigma_{x,i}^2\right\}_{i=1}^m$, the dictionary A and the noise variances σ_e^2 are known. For a general dictionary A and an arbitrary symmetric interaction matrix W the exact MAP estimator requires an exhaustive search over all 2^m possible supports. To overcome the infeasibility of the combinatorial search, two approaches can be taken. The first is to develop an efficient approximation of the MAP estimator. We develop such an algorithm in Section V. An alternative strategy is to make additional assumptions on the model parameters, namely on A and W, that will make exact MAP estimation feasible. This is addressed in Section VI, where we consider unitary dictionaries A and decomposable BMs. The more practical setup where the model parameters are also unknown is considered in Section VIII, for which we derive efficient methods for estimating both the sparse representations and the model parameters from a set of signals.

V. Approximate MAP estimation

As we have seen in the previous section, exact MAP estimation requires an exhaustive search over all 2^m possible supports. To simplify the computations, we propose a greedy approximation. We begin by explaining the core idea of our greedy algorithm. Our goal is to estimate the support which achieves the maximal value of the posterior probability $\Pr(S|y)$. This means that our objective function is the one that appears in (10). We start with an empty support, which means that $\{S_i\}_{i=1}^m$ are all -1. At the first iteration, we check each of the m possible elements that can be added to the empty support and evaluate the term in (10). The entry i_* leading to the largest value is chosen and thus S_{i_*} is set to be $+1$. Given the updated support, we proceed exactly in the same manner. In every iteration we consider all the remaining inactive elements and choose the one that leads to the maximal value in (10) when added to the previously set support. The

algorithm stops when the value of (10) is decreased for every additional item in the support.

In each iteration only one entry in S changes - from -1 to 1. This can be used to simplify some of the terms that appear in (10):

$$\begin{aligned}\frac{1}{2}S^T W S &= \frac{1}{2}\sum_{i,j} W_{ij}S_i S_j = C_1 + 2\sum_j W_{ij}S_j \\ b^T S &= \sum_{i=1}^m b_i S_i = C_2 + 2b_i \\ \sum_{i=1}^m \ln\left(\sigma_{x,i}^2/\sigma_e^2\right) S_i &= C_3 + 2\ln\left(\sigma_{x,i}^2\right)\end{aligned} \tag{12}$$

where C_1, C_2, C_3 are constants that will not be needed in our derivation. Consequently, in each iteration it is sufficient to find an index i (out of the remaining inactive indexes) that maximizes the following expression:

$$\begin{aligned}Val(i) =& \frac{1}{2\sigma_e^2} y^T A_{s^k} Q_{s^k}^{-1} A_{s^k}^T y - \frac{1}{2}\ln\left(\left|\det\left(Q_{s^k}\right)\right|\right) + \\ & 2W_i^T S^k + 2b_i - \frac{1}{2}\ln\left(\sigma_{x,i}^2\right)\end{aligned} \tag{13}$$

where s_k is the support estimated in iteration $k-1$ with the entry i added to it, $Q_{s^k} = A_{s^k}^T A_{s^k} + \sigma_e^2 \Sigma_{s^k}^{-1}$ and W_i^T is the ith row of W. A pseudo-code for the proposed greedy algorithm is given in Algorithm 1.

Algorithm 1 Greedy algorithm for approximating the MAP estimator of (10)

Input: Noisy observations $y \in \mathbb{R}^n$ and model parameters $W, b, \{\sigma_{x,i}\}_{i=1}^m, A, \sigma_e$.
Output: A recovery $\hat{s}_{\text{MAP}}$ for the support.

```
s_*^0 = ∅,  S_*^0 = −1^{m×1}
k = 1
repeat
  for i ∉ s_*^{k−1} do
    s^k = s_*^{k−1} ∪ i
    S^k[j] = { S^{k−1}[j]   ,  j ≠ i
             { 1            ,  j = i
    Evaluate Val(i) using (13).
  end for
  i_* = argmax_i {Val(i)}
  s_*^k = s_*^{k−1} ∪ i_*,  S_*^k[j] = { S_*^{k−1}[j]  ,  j ≠ i_*
                                       { 1             ,  j = i_*
  k = k + 1
until Pr(s_*^k|y) < Pr(s_*^{k−1}|y)
Return: ŝ_MAP = s_*^{k−1}
```

We now provide some intuitive meaning to the terms in (13). The term $y^T A_{s^k} Q_{s^k}^{-1} A_{s^k}^T y$ is equivalent to the residual error $\left\|r^k\right\|_2^2$, where $r^k = y - A_{s^k}\left(A_{s^k}^T A_{s^k}\right)^{-1} A_{s^k}^T y$ is the residual in respect to the signal. To see that, notice that the following relation holds:

$$\left\|r^k\right\|_2^2 = \|y\|_2^2 - y^T A_{s^k}\left(A_{s^k}^T A_{s^k}\right)^{-1} A_{s^k}^T y. \tag{14}$$

Using the definition of Q_{s^k} it can be easily verified that the two terms take a similar form, up to a regularization factor in the pseudoinverse of A_{s^k}. Next, we turn to the terms $W_i^T S^k$ and b_i. The first corresponds to the sum of interactions between the ith atom and the rest of the atoms which arise from turning it on (the rest remain unchanged). The second term is the separate bias for the ith atom. As the sum of interactions and the separate bias become larger, using the ith atom for the representation leads to an increase in the objective function. Consequently, the total objective of (13) takes into consideration both the residual error in respect to the signal and the prior on the support. This can lead to improved performance over greedy pursuit algorithms like OMP and CoSaMP, which are aimed at minimizing the residual error alone.

To conclude this section, note that the recent work of [23] used a BM-based Bayesian modeling for the sparse representation to improve the CoSaMP algorithm. The resulting algorithm is referred as lattice matching pursuit (LaMP). The inherent differences between our approach and the one suggested in [23] are explained in Section IX.

VI. Exact MAP estimation

A. Model Assumptions

In this section we consider a simplified setup where exact MAP estimation is feasible. A recent work [15] treated the special case of a unitary dictionary for independent-based priors, and developed closed-form expressions for the MAP and MMSE estimators. We follow a similar route here and assume that the dictionary is unitary. In this case we can make a very useful observation which is stated in Theorem 1. A proof of this theorem is provided in Appendix A.

Theorem 1: Let A be a unitary dictionary. Then the BM distribution is a conjugate prior for the MAP estimation problem of (10), namely the *a posteriori* distribution $\Pr(S|y)$ is a BM with the same interaction matrix W and a modified bias vector q with entries:

$$q_i = b_i + \frac{1}{4}\left\{\frac{\sigma_{x,i}^2}{\sigma_e^2(\sigma_e^2+\sigma_{x,i}^2)}\left(y^T a_i\right)^2 - \ln\left[1+\frac{\sigma_{x,i}^2}{\sigma_e^2}\right]\right\} \tag{15}$$

for all i, where a_i is the ith column of A.

Notice in (15) that q_i is linearly dependent on the original bias b_i and quadratically dependent on the inner product between the signal y and the atom a_i. This aligns with the simple intuition that an atom is more likely to be used for representing a signal if it has an *a priori* tendency to be turned "on" and if it bares high similarity to the signal (this is expressed by a large inner product). From Theorem 1 the MAP estimation problem of (10) takes on the form of integer programming. More specifically, this is a Boolean quadratic program (QP):

$$\underset{S}{\text{maximize}}\left(q^T S + \frac{1}{2}S^T W S\right) \text{ s.t. } S_i^2 = 1,\ 1 \le i \le m. \tag{16}$$

This is a well-known combinatorial optimization problem [27] that is closely related to multiuser detection in communication systems, a long-studied topic [28]. The Boolean QP remains computationally intensive if we do not use any approximations

or make any additional assumptions regarding the interaction matrix W. The vast range of approximation methods used for multiuser detection, like semi-definite programming (SDP) relaxation, can be adapted to our setup. Another approximation approach, which is commonly used for energy minimization in the BM, is based on a Gibbs sampler and simulated annealing techniques [17], which remain computationally demanding. Our interest here is in cases for which simple exact solutions exist. We therefore relax the dependency model, namely make additional modeling assumptions on W.

We first consider the simple case of $W = 0$, which corresponds to the independency assumption. Using Theorem 1, we can follow the same analysis as in Section III-B for $W = 0$ by replacing the bias vector b by q. Consequently, in this case we have:

$$\Pr(S|y) = \prod_{i=1}^{m} \Pr(S_i|y), \tag{17}$$

where $\Pr(S_i = 1|y) = {}^{1}/_{(1+\exp(-2q_i))}$ for all i. Notice that $\Pr(S_i = 1|y) > \Pr(S_i = -1|y)$ if $q_i > 0$. This means that the ith entry of $\hat{S}_{MAP}$ equals 1, namely i is in the support, if $q_i > 0$. Using (15) we obtain the following MAP estimator for S:

$$\hat{S}_{i,MAP} = \begin{cases} 1, & |y^T a_i| > \frac{\sqrt{2}\sigma_e}{c_i}\sqrt{\ln\left[\frac{1-p_i}{\sqrt{1-c_i^2}p_i}\right]} \\ -1, & otherwise \end{cases} \tag{18}$$

where p_i is defined in (4) and $c_i = \sqrt{\sigma_{x,i}^2/(\sigma_{x,i}^2+\sigma_e^2)}$. These results correspond to those of [15] for the MAP estimator under a unitary dictionary.

To add dependencies into our model, we may consider two approaches, each relying on a different assumption on W. First, we can assume that all entries in W are non-negative. If this assumption holds, then the energy function defined by the Boltzmann parameters $W,\ q$ is regarded "sub-modular" and it can be minimized via graph cuts [25]. The basic technique is to construct a specialized graph for the energy function to be minimized such that the minimum cut on the graph also minimizes the energy. The minimum cut, in turn, can be computed by max flow algorithms with complexity which is polynomial in m. The recent work [23] is based on this approach and we will relate to it in more detail in Section IX.

Here we take a different approach, which seems to be more appropriate for our setup. This approach makes an assumption on the structural component of the MRF - we assume that the BM is **decomposable with a small tree-width**. This type of MRF was explored in detail in Section III-C. The above assumption implies that the matrix W has a special sparse structure - it corresponds to a chordal graph where the size of the largest maximal clique is small. As we have seen in Section III-C, decomposable models can serve as a very useful relaxation for general dependency models. Another motivation for this assumption arises from the results that were shown in Section II for the special case of image patches and a DCT dictionary. It was shown there that independency can be considered a reasonable assumption for many pairs of DCT atoms. This observation has the interpretation of a sparse structure for the interaction matrix W. Consequently, it seems plausible that a matrix W with a sparse structure can capture most of the significant interactions in this case.

From Theorem 1 it follows that if the above assumption on the structure of W holds for the BM prior on S it also holds for BM posterior (since both distributions correspond to the same interaction matrix). We can therefore use belief propagation techniques to find the MAP solution. We next present in detail a concrete message passing algorithm for obtaining an exact solution to (16) under a banded W matrix.

B. The Message-Passing Algorithm

Before we go into the details of the proposed message-passing algorithm, we make a simple observation that will simplify the formulation of this algorithm. As we have seen in Section III-B, a posterior BM distribution with parameters W, q can be written (up to a normalization factor which has no significance in the MAP estimation problem) as a product of potential functions defined on the maximal cliques in the corresponding graph:

$$\exp\left(q^T S + \frac{1}{2}S^T W S\right) = \prod_{i=1}^{P} \Psi_{C_i}(S_{C_i}) \tag{19}$$

where P is the the number of maximal cliques. By replacing the potentials $\{\Psi_{C_i}(S_{C_i})\}$ with their logarithms, which are denoted by $\left\{\widetilde{\Psi}_{C_i}(S_{C_i})\right\}$, we remain with quadratic functions of the variables of $\{S_i\}_{i=1}^{m}$:

$$S^T W S + q^T S = \sum_{i=1}^{P} \widetilde{\Psi}_{C_i}(S_{C_i}). \tag{20}$$

This can be very useful from a computational point of view as there is no need to compute exponents, which can lead to large values. Each product that appears in a standard message-passing algorithm is replaced by summation.

For concreteness we will focus on the special case of an Lth order banded interaction matrix W of size m-by-m, as described in Section III-C. In this case the maximal cliques are $C_i = \{S_i, \ldots, S_{i+L}\},\ i = 1, \ldots, m-L$, so that all cliques are of size $L+1$ and the tree-width is L. The clique tree takes the form of a simple chain of length $m - L$. We denote the "innermost" clique in this chain by C_k, where $k = \left\lceil \frac{m-L-1}{2} \right\rceil$. We choose an order for the cliques where the cliques at both edges of the chain appear first and the "innermost" clique appears last and set the clique potentials according to the rule of thumb that was mentioned in Section III-B. Consequently, the logarithms of the potentials are given by:

$$\widetilde{\Psi}_{C_i} = \begin{cases} q_i S_i + \sum_{l=i+1}^{i+L} W_{il} S_i S_l & ,\ 1 \le i \le k-1 \\ \sum_{j=k}^{k+L} q_j S_j + \sum_{j=k}^{k+L-1} \sum_{l=j+1}^{k+L} W_{jl} S_j S_l & ,\ i = k \\ q_{i+L} S_{i+L} + \sum_{l=i}^{i+L-1} W_{l,i+L} S_l S_{i+L} & ,\ k+1 \le i \le m-L \end{cases} \tag{21}$$

$\widetilde{\Psi}_{C_i}$ is a function of $S_i, \ldots, S_{i+L}$. We pass messages "inwards" starting from C_1 and C_{m-L} until the clique C_k

receives messages from both sides:

$$m_{i,i+1} = \begin{cases} \max\limits_{S_i} \widetilde{\Psi}_{C_i} & , \ i = 1 \\ \max\limits_{S_i} \widetilde{\Psi}_{C_i} + m_{i-1,i} & , \ 2 \leq i \leq k-1 \end{cases} \quad (22)$$

$$m_{i,i-1} = \begin{cases} \max\limits_{S_{i+L}} \widetilde{\Psi}_{C_i} & , \ i = m-L \\ \max\limits_{S_{i+L}} \widetilde{\Psi}_{C_i} + m_{i+1,i} & , \ m-L-1 \leq i \leq k+1 \end{cases}$$

where $m_{i,i+1}$ depends on $S_{i+1}, \ldots, S_{i+L}$ and $m_{i,i-1}$ on $S_i, \ldots, S_{i+L-1}$. The arguments that correspond to each of the maximization operators are denoted by $\Phi_{i,i+1},\ i = 1, \ldots, k-1$ and $\Phi_{i,i-1},\ i = k+1, \ldots, m-L$ (these have the same form as the messages with "max" replaced by "argmax"). The MAP estimates are then computed recursively by:

$$\begin{aligned} \left(S_k^*, \ldots, S_{k+L}^*\right) &= \underset{S_k, \ldots, S_{k+L}}{\text{argmax}} \ \widetilde{\Psi}_{C_k} + m_{k-1,k} + m_{k+1,k} \\ S_i^* &= \Phi_{i,i+1}\left(S_{i+1}^*, \ldots, S_{i+L}^*\right),\ i = k-1, \ldots, 1 \\ S_{i+L}^* &= \Phi_{i,i-1}\left(S_i^*, \ldots, S_{i+L-1}^*\right),\ i = k+1, \ldots, m-L. \end{aligned} \quad (23)$$

The message-passing algorithm in this case is summarized in Algorithm 2.

Algorithm 2 Message-passing algorithm for obtaining the exact MAP estimator of (10) in the special case of a unitary dictionary and a banded interaction matrix

Input: Noisy observations y and model parameters $W, b, \{\sigma_{x,i}\}_{i=1}^m, A, \sigma_e$. A is unitary and W is an Lth order banded matrix.

Output: A recovery $\hat{S}_{\text{MAP}}$ for the sparsity pattern of x.

Step 1: Set the bias vector q for the BM posterior distribution $\Pr(S|y)$ using (15).

Step 2: Assign a potential function $\widetilde{\Psi}_{C_i}\left(S_{C_i}\right)$ for each clique $C_i = \{S_i, \ldots, S_{i+L}\},\ i = 1, \ldots, m-L$ using (21).

Step 3: Pass messages "inwards" starting from C_1 and C_{m-L} until the "innermost" clique C_k receives messages from both sides using (22).

Step 4: Obtain the MAP estimate for S using (23).

An important observation is that the complexity of the proposed algorithm is exponential in L and not in m. More specifically the complexity is $O(2^L \cdot m)$. As the value of L is part of our modeling, even when m is relatively large (and the exhaustive search which depends on 2^m is clearly infeasible), the exact MAP computation is still feasible as long as L remains sufficiently small. If we have for example $L = \gamma \log_2(m)$ then the complexity is $O(m^{1+\gamma})$, namely it is polynomial in m.

VII. Simulations on Synthetic Signals

In this section we test the two recovery algorithms that were proposed in the two previous sections (see Algorithms 1,2) and compare their performance to that of previous sparse recovery methods. We assume here that all the parameters of the BM-based generative model are known and use this model to create random data sets of signals, along with their sparse representations. A standard Gibbs sampler [17] is used for sampling sparsity patterns from the BM. The sampled supports and representation vectors are denoted by $\left\{s^{(l)}, x^{(l)}\right\}_{l=1}^N$.

We begin by examining a setup that satisfies the simplifying assumptions of Section VI. We assume that the dictionary $A \in \mathbb{R}^{m \times m}$ is a unitary DCT dictionary with $m = 64$, and that W is a banded interaction matrix with $L = 9$. The nonzero entries in the upper triangle of W are drawn independently from $\mathcal{U}[-\Delta_W, \Delta_W]$ (the lower triangle is determined from the symmetry of W) and the entries in the bias vector $b \in \mathbb{R}^m$ are drawn independently from $\mathcal{N}(b_0, 1)$. The parameters $\{\sigma_{x,i}\}_{i=1}^m$ are in the range $[15, 60]$. In this case we can apply both of the algorithms that were suggested in this paper. We also consider two additional algorithms - a standard pursuit algorithm like OMP and a MAP recovery that is based on an independent-based prior like the one that appears in (18). The OMP algorithm is used only for identifying the support. Then the recovered support is used to obtain an estimate for the representation vector using (11), just as the MAP estimators. Note that the marginal probabilities $\{p_i\}_{i=1}^m$ for (18) are computed from the Boltzmann parameters using standard belief propagation techniques (see III-C). We compare the performance of the four algorithms for different noise levels - σ_e is in the range $[2, 30]$.

In order to explore the dependency of the recovery algorithms on the Boltzmann parameters, we create different data sets, each consisting of $N = 10,000$ signals and corresponding to different values of Δ_W and b_0. For each of the above-mentioned algorithms we evaluate two performance criteria. The first one is the probability of error in identifying the true support:

$$1 - \frac{1}{N} \sum_{l=1}^{N} \frac{|s^{(l)} \cap \hat{s}^{(l)}|}{\max(|s|, |\hat{s}|)}. \quad (24)$$

The second criterion is the relative recovery error, namely the mean recovery error for the representation coefficients normalized by their energy:

$$\sqrt{\frac{\sum\limits_{l=1}^{N} \|\hat{x}^{(l)} - x^{(l)}\|_2^2}{\sum\limits_{l=1}^{N} \|x^{(l)}\|_2^2}}. \quad (25)$$

The relative error is also evaluated for the Bayesian oracle estimator, namely the oracle which knows the true support. Note that for a unitary dictionary the relative error for the representation coefficients is in fact also the relative error for the noise-free signal, since $\|Au\|_2^2 = \|u\|_2^2$ for any vector u. We performed experiments for a wide range of data sets. However, for concreteness, we show only several results in Fig. 4. The MAP estimator of (18) is denoted in the figures by "MAP - independency".

The results in Fig. 4 show that all three MAP estimators outperform the OMP algorithm, both in terms of the recovered support and the recovery error. For $\Delta_W = 0.5$ the greedy MAP and the independent-based MAP serve as excellent approximations for the exact BM-based MAP. As we turn to stronger interactions - the variance of b remains the same,

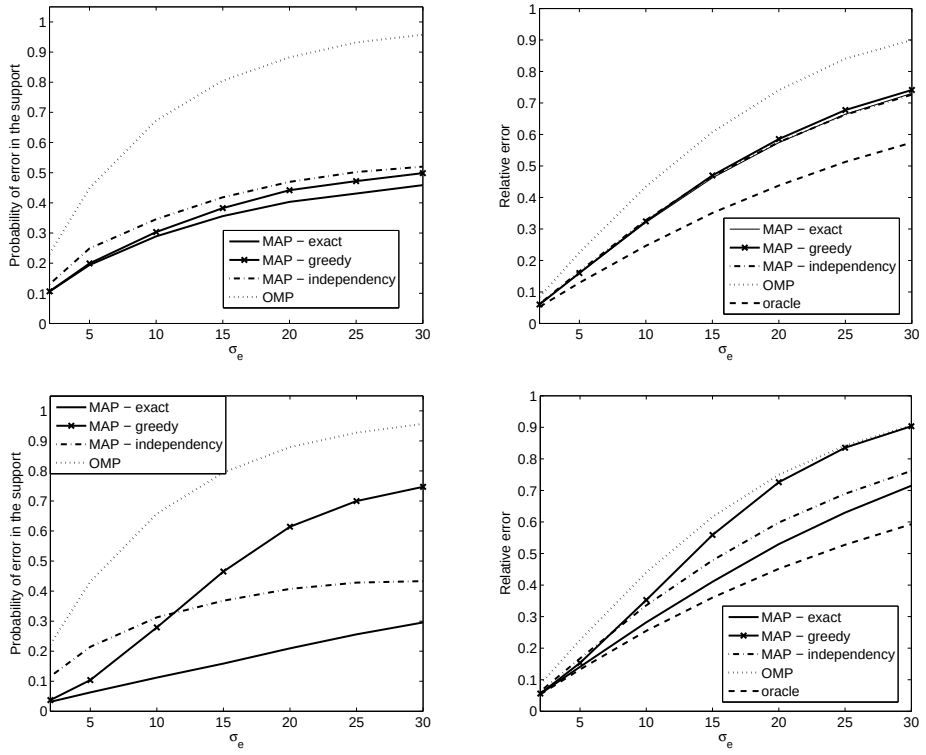


Figure 4. Probability of error in identifying the support (24) and relative recovery error (25) for a unitary DCT dictionary and a banded interaction matrix. Top: A data set with $b_0 = -1.5$ and $\Delta_W = 0.5$, leading to an empirical average cardinality $|s| = 11$. Bottom: A data set with $b_0 = -3.5$ and $\Delta_W = 2$, leading to an empirical average cardinality $|s| = 13$.

while Δ_W is increased from 0.5 to 2 - the quality of these approximations deteriorates. In this case b_0 is decreased from -1.5 to -3.5, so that the average cardinality remains more or less the same as before. For "strong" interactions, the two MAP approximations exhibit a different behavior. At low noise levels ($\sigma_e = 5$ and below) the greedy algorithm serves as a very good approximation to the exact MAP, whereas for higher noise levels the performance gap increases rapidly. The independent-based MAP however is a bit less accurate than the greedy one at noise levels below $\sigma_e = 10$, but it closes up the performance gap in respect to the exact MAP as the noise level increases.

We now provide some additional observations that were drawn from similar sets of experiments which are not shown here. We observed that increasing Δ_W without changing b_0 leads to performance gap described above. However, when b_0 is increased and Δ_W remains unchanged, the approximations for MAP align with the exact estimator. Note that in both cases the average cardinality is increased. We can conclude from these observations and from the results that appear in Fig. 4 that the performance gap results from the increase in the interaction level and not from the increase in the cardinalities. As for higher noise levels, we noticed that all algorithms exhibit saturation in their performance. In this setup the OMP tends to choose an empty support, leading to an obvious failure in its recovery. Another interesting observation is that the independent-based MAP aligns with the exact MAP for high noise levels (above $\sigma_e = 50$).

Next, we turn to the case of a redundant dictionary and a general (non-sparse) interaction matrix. We use an over-complete 64-by-256 DCT dictionary. All the rest of model parameters are the same as before, expect for the interaction matrix which is no longer banded (we use the same distribution as before for all the entries in the upper triangle). For this setup exact MAP estimation is no longer possible and we can use only the greedy approximation for MAP. We compare the performance of our BM-based greedy algorithm to that of OMP and a greedy approximation for an independent-based MAP. In the latter we use Algorithm 1 with $W = 0$ and $b_i = \frac{1}{2}\ln\left({}^{p_i}/_{(1-p_i)}\right),\ i = 1,\ldots,m$, where the single node probabilities $p_i = \Pr\left(S_i = 1\right)$ are computed empirically from the data. In this setup we evaluate the probability of error in the support (24) and the relative recovery error in respect to the noise-free signal:

$$\sqrt{\frac{\sum_{l=1}^{N} \|A\hat{x}^{(l)} - Ax^{(l)}\|_2^2}{\sum_{l=1}^{N} \|Ax^{(l)}\|_2^2}}. \tag{26}$$

The results are shown in Fig. 5. We see that both greedy approximations of MAP clearly outperform the OMP algorithm and that the greedy approximation which takes into consideration the interactions is superior over the one which ignores them.

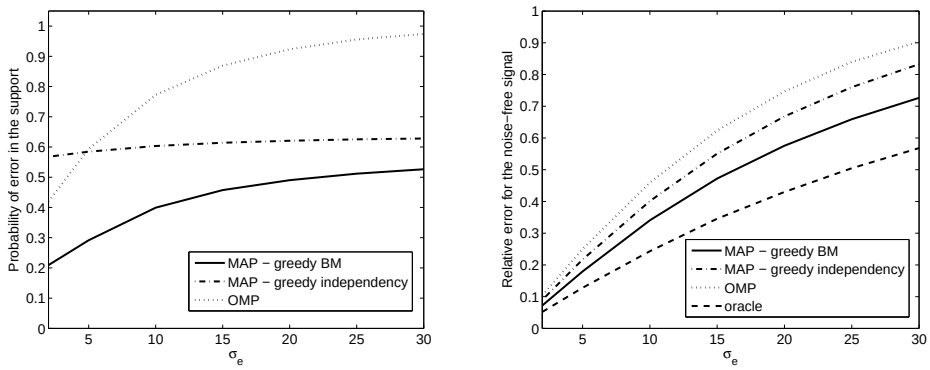


Figure 5. Probability of error in identifying the support (24) and relative recovery error in respect to the noise-free signal (26) for an overcomplete 64-by-256 DCT dictionary and a general (non-sparse) interaction matrix. The entries in W are drawn independently from $\mathcal{U}[-0.1, 0.1]$ and the entries in the bias vector $b \in \mathbb{R}^m$ are drawn independently from $\mathcal{N}(-2.5, 1)$. This leads to an empirical average cardinality $|s| = 14.5$.

VIII. Adaptive Sparse Signal Recovery

In an actual problem suite we are given a set of signals $\left\{y^{(l)}\right\}_{l=1}^{N}$ from which we would like to estimate both the sparse representations and the model parameters. We briefly address this joint estimation problem in this section. For concreteness we focus here only on the estimation of the Boltzmann parameters. We begin by assuming that we are given the sparse representations, namely we have a data set of i.i.d. examples $\mathcal{D} = \left\{y^{(l)}, x^{(l)}, S^{(l)}\right\}_{l=1}^{N}$, from which we would like to learn the Boltzmann parameters W, b. We consider a maximum likelihood (ML) approach for estimating W, b:

$$\left[\hat{W}_{ML}, \hat{b}_{ML}\right] = \underset{W,b}{\operatorname{argmax}}\ \mathcal{L}(W, b) \tag{27}$$

where

$$\mathcal{L}(W, b) = \frac{1}{2}\sum_{l=1}^{N}\left[\left(S^{(l)}\right)^T W S^{(l)} + b^T S^{(l)}\right] - N\ln(Z(W, b)) \tag{28}$$

is the log likelihood function for the Boltzmann parameters, namely $\mathcal{L}(W, b) = \ln\left(\Pr\left(\mathcal{D}|W, b\right)\right)$. We can see from (27) that the estimation of W, b depends only on the supports $\left\{S^{(l)}\right\}_{l=1}^{N}$.

ML estimation of W, b is computationally intensive due to the exponential complexity in m associated with the partition function $Z(W, b)$. Therefore, we turn to approximated ML estimators. A widely used approach is applying Gibbs sampling and mean-field techniques, see for example [22]. However, these methods are usually computationally demanding. A simpler approach is to replace the likelihood function by pseudo-likelihood (PL), leading to MPL estimation. This approach was presented in [29] and revisited in [30], where it was shown that the MPL estimator is consistent. This means that in the limit of infinite sampling ($N \to \infty$), the PL function is maximized by the true parameter values.

The basic idea in MPL estimation is to replace the BM prior $\Pr(S|W, b)$ by the product of all the conditional distributions of each node S_i given the rest of the nodes S_{ic}: $\prod_{i=1}^{m}\Pr\left(S_i|S_{ic}, W, b\right)$. Each of these conditional distributions takes on the simple form

$$\Pr\left(S_i|S_{ic}, W, b\right) = C\exp\left\{S_i\left(W_i^T S + b_i\right)\right\} \tag{29}$$

where W_i^T is the ith row of W and C is a normalization constant. Since this is a probability distribution for a single binary node S_i it follows that $C = 2\cosh\left(W_i^T S + b_i\right)$. Consequently, we replace $\Pr(S|W, b)$ by

$$\prod_{i=1}^{m}\Pr\left(S_i|S_{ic}, W, b\right) = \prod_{i=1}^{m}\frac{\exp\left\{S_i\left(W_i^T S + b_i\right)\right\}}{2\cosh\left(W_i^T S + b_i\right)} = \frac{\exp\left\{S^T\left(WS + b\right)\right\}}{2^m\prod_{i=1}^{m}\cosh\left(W_i^T S + b_i\right)}. \tag{30}$$

We define the log-PL by:

$$\mathcal{L}_p(W, b) = \sum_{l=1}^{N}\sum_{i=1}^{m}\ln\left(\Pr\left(S_i^{(l)}|S_{ic}^{(l)}, W, b\right)\right) \tag{31}$$

$$= \sum_{l=1}^{N}\left(S^{(l)}\right)^T\left(WS^{(l)} + b\right) - \mathbf{1}^T\rho\left(WS^{(l)} + b\right) - mN\ln(2)$$

where $\rho(z) = \ln(\cosh(z))$ and the function $\rho(\cdot)$ operates on a vector entry-wise. To explore the properties of the log-PL function it is useful to place all the Boltzmann parameters - there are $p = {}^{(m^2+m)}/_2$ unknowns (${}^{(m^2-m)}/_2$ in the upper triangle of W and m in b) - in a column vector u. For each example $S^{(l)}$ in the data set we can construct matrices $B^{(l)}$, $C^{(l)}$ so that $B^{(l)}u = \left(S^{(l)}\right)^T\left(WS^{(l)} + b\right)$ and $C^{(l)}u = WS^{(l)} + b$.

Using these notations the log-PL function of (31) can be re-formulated as:

$$\mathcal{L}_p(u) = \sum_{l=1}^{N}\left[B^{(l)}u - \mathbf{1}^T\rho\left(C^{(l)}u\right)\right] - mN\ln(2). \tag{32}$$

The gradient and the hessian of $\mathcal{L}_p(u)$ are given by:

$$\nabla\mathcal{L}_p(u) = \sum_{l=1}^{N}\left[\left(B^{(l)}\right)^T - \left(C^{(l)}\right)^T\rho'\left(C^{(l)}u\right)\right] \tag{33}$$

$$\nabla^2\mathcal{L}_p(u) = -\sum_{l=1}^{N}\left[\left(C^{(l)}\right)^T\operatorname{diag}\left(\rho''\left(C^{(l)}u\right)\right)C^{(l)}\right], \tag{34}$$

where $\rho'(z) = \tanh(z)$ and $\rho''(z) = 1 - \tanh^2(z)$. Since $\rho(z)$ is a convex function, it follows that the log-PL function is concave in u. Therefore, as an unconstrained convex optimization, we have many reliable algorithms that could be of use.

In [30] MPL estimation is treated by means of gradient ascent (GA) methods. These methods are very simple, but it is well-known that they suffer from a slow convergence rate [31]. Another optimization algorithm which converges more quickly is Newton [31]. Note however that the problem dimensions here can be very large. For example, when $m = 64$ as in an 8-by-8 image patch, we have $p = 2080$ unknown parameters. Since Newton iterations requires inverting the Hessian matrix, it becomes computationally demanding. Instead we would like to use an efficient algorithm that can treat large-scale problems. To this end we suggest the sequential subspace optimization (SESOP) method [32], which is known to lead to a significant speedup in respect to gradient descent.

The basic idea in SESOP is to use the following update rule for the parameter vector in each iteration:

$$u^{j+1} = u^j + Q^j v^j, \tag{35}$$

where Q^j is a matrix consisting of various (normalized) direction vectors in its columns and v^j is a vector containing the step size in each direction. In our setting we use only the current gradient $g^j = \nabla \mathcal{L}_p(u^j)$ and M recent steps $p^i = u^i - u^{i-1},\ i = j-M, \ldots, j-1$. We use the abbreviation SESOP-M for this mode of the algorithm. The vector v^k is determined in each iteration by an inner optimization stage. Since we use a small number of directions, the optimization problem in respect to v^j is a small-scale one and we can apply Newton iterations to solve it, using $\nabla_{v^j} \mathcal{L}_p(u^{j+1}) = \left(Q^j\right)^T \nabla \mathcal{L}_p(u^{j+1})$ and $\nabla^2_{v^j} \mathcal{L}_p(u^{j+1}) = \left(Q^j\right)^T \nabla^2 \mathcal{L}_p(u^{j+1}) Q^j$.

To initialize the algorithm we set the interaction matrix to zero, namely we allow no interactions. We then perform a separate MPL estimation of b where W is fixed to zero, which results in

$$\hat{b}_i^0 = \text{atanh}\left[\frac{1}{N}\sum_{l=1}^{N} S_i^{(l)}\right], \tag{36}$$

for all i. We stop the algorithm either when the norm of the gradient vector $\nabla \mathcal{L}_p(u)$ decreases below a pre-determined threshold ϵ, or after a fixed number of iterations J. A pseudo-code that summarizes the learning algorithm for the Boltzmann parameters is provided in Algorithm 3.

To demonstrate the effectiveness of MPL estimation via SESOP, we now show some results of synthetic simulations. We focus on a 9th order banded interaction matrix of size 64-by-64 and follow the same setup we used in Section VII for the simulations on the unitary dictionary, with parameters $\Delta_W = 0.5,\ b_0 = -1.5$ and a data set of size $N = 16,000$. We use the true support vectors, produced by the Gibbs sampler, as an input for the learning algorithm and apply 50 iterations of both GA and SESOP-2 to estimate the Boltzmann parameters. The results are shown in Fig. 6. We can see on the top that SESOP outperforms GA both in terms of convergence rate of the PL objective and recovery error for the interaction matrix. This is also demonstrated visually on the middle and bottom, where we can see that for the same number of iterations SESOP reveals much more interactions than GA. In fact, if we set to zero the entries in the true W that correspond to rarely used atoms (i.e. if the appearance frequency of atoms i or j is very low then we set $W_{ij} = 0$), we can see that SESOP was able to learn most of the significant interactions.

Algorithm 3 A SESOP-M algorithm for obtaining the MPL estimator of the Boltzmann parameters

Input: A data set of supports $\left\{S^{(l)}\right\}_{l=1}^{N}$.
Output: A recovery $\hat{W}, \hat{b}$ for the Boltzmann parameters.
Initialization: Set $\hat{W}$ to zero and $\hat{b}^0$ according to (36), and construct from them a column vector $\hat{u}^0$.
$j = 0$
repeat
Step 1: Evaluate $\mathcal{L}_p(\hat{u}^j)$ and $\nabla \mathcal{L}_p(\hat{u}^j)$ using (32)-(33).
Step 2: Set the matrix Q^j using the current gradient $\nabla \mathcal{L}_p(\hat{u}^j)$ and M previous steps $\left\{\hat{u}^i - \hat{u}^{i-1}\right\}_{i=j-M}^{j-1}$.
Step 3: Determine the step sizes v^j by Newton iterations.
Step 4: $\hat{u}^{j+1} = \hat{u}^j + Q^j v^j$.
$j = j + 1$
until $\nabla \mathcal{L}_p(\hat{u}^j) < \epsilon$ or $j \geq J$
Return: $\hat{W}$, $\hat{b}$ extracted out of $\hat{u}^j$.

So far we focused on estimating W, b. However, given the data set, we often need to evaluate **all** the model parameters, including the dictionary A and the variances $\left\{\sigma^2_{x_i}\right\}_{i=1}^{m}$. Furthermore, in practice the sparse representations are also unknown. We suggest using a block-coordinate optimization approach for approximating the solution of the joint estimation problem, which results in an iterative scheme for adaptive sparse signal recovery. Each iteration in this scheme consists of two stages: sparse coding where we apply a MAP estimator for the sparse representations when the model parameters are fixed, and model update based on the current estimate of the sparse representations. First steps towards this goal are taken in [33], where we demonstrate the effectiveness of the adaptive model-based approach on image patches. We intend to explore this further in our future work.

IX. Relation to past works

In this section we briefly relate to two recent works [22], [23] that used the BM as a prior on the support of the representation vector. We discuss their main contributions and drawbacks, and emphasize the differences in our work with respect to them. In recent years capturing and exploiting dependencies between dictionary atoms has become a hot topic in the model-based sparse recovery field. In contrast to previous works like [3], [7], [20], [21] which considered dependencies in the form of tree structures, [22] was the first to propose a general and adaptive model for capturing these dependencies. The main contribution of this work is the proposal of a new sparse coding model, which is represented by a graphical model. The model is based on the celebrated BM prior, and is provided with a biological motivation through the architecture of the visual cortex. Note that we used exactly the same graphical model in our work (see Section IV).

In [22] MAP estimation of the sparse representation and learning of the Boltzmann parameters are handled by means of general-purpose optimization techniques. For MAP estimation [22] proposes Gibbs sampling and simulated annealing, and

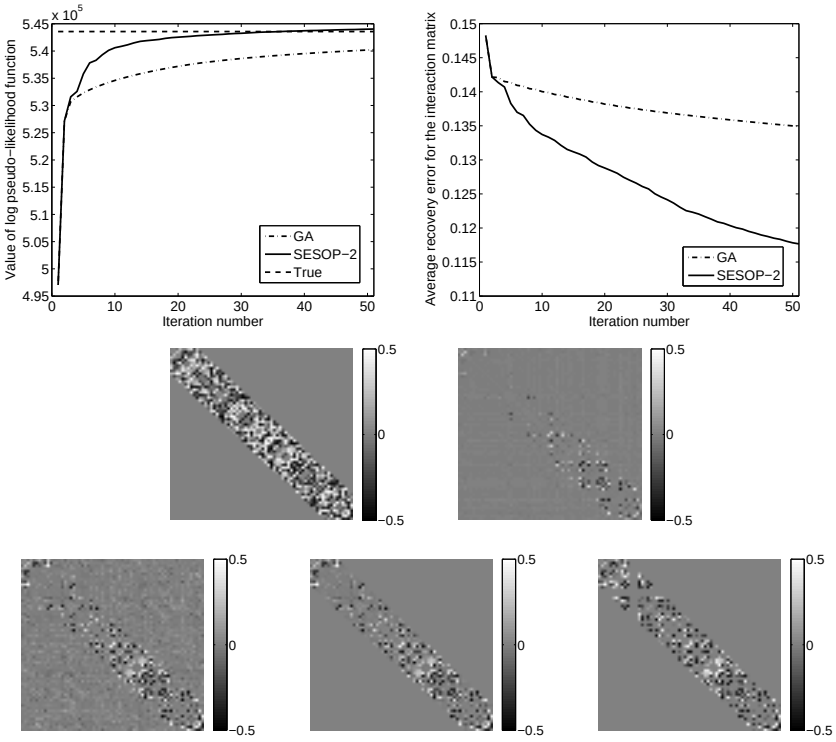


Figure 6. Top - results of MPL estimation via GA and SESOP: The value of the log-PL objective and the average recovery error for the interaction matrix per entry as functions of the number of iterations. Middle (from left to right): The true interaction matrix W and MPL estimate via GA $\hat{W}_{GA}$. Bottom (from left to right): MPL estimate via SESOP $\hat{W}_{SESOP}$, a banded version of it and a matrix consisting of the interactions in W which are more likely to be revealed using the given data set. We can see that the latter two are very close.

for learning the Boltzmann parameters they suggest Gibbs sampling and mean-field approximations. Note that these techniques require a high computational effort and suffer from a slow convergence rate. In fact, as the main purpose of [22] is to introduce the concept of interactions in a sparse coding model, little effort was invested into algorithmic design. In this sense, our work serves as a natural extension to [22]. We develop specialized algorithms for both MAP estimation and learning of the Boltzmann parameters, which are efficient and reliable and at the same time still quite simple.

Next, we turn to [23]. This work adapts a signal model like the one presented in [22], with several modifications. First, it is assumed that all the weights in the interaction matrix W are nonnegative. Second, the Gaussian distributions for the nonzero representation coefficients are replaced by parametric utility functions. The main contribution of [23] is using the BM generative model for extending the CoSaMP algorithm, a well known greedy method. The extended algorithm, referred as LaMP, differs from CoSaMP in the stage of the support update in each iteration, which becomes more accurate. This stage is now based on graph cuts and this calls to the nonnegativity constraint on the entries of W. The rest of the iterative scheme however remains unchanged and is still based on "residuals": in each iteration we compute the residual with respect to the signal and the algorithm stops when the residual error falls below a pre-determined threshold. Note that LaMP requires the desired sparsity level as an input, just like CoSaMP.

In our work we take a different greedy approach and use the Bayesian framework to its full extent. The BM-based generative model is incorporated into all of the stages of the greedy algorithm, including its stopping rule. Our greedy algorithm works for an arbitrary interaction matrix and in this sense it is more general than LaMP. Note also that there is no need to provide our algorithm with the desired sparsity level. Due to the inherent differences between the setups that are addressed by each of the greedy algorithms, we did not compare their denoising performance. Further work is required in order to integrate our Bayesian setting into a CoSaMP-like algorithm and then a more meaningful test could be performed.

X. Conclusions

In this work we have developed a scheme for adaptive model-based recovery of sparse representations. We adapted a Bayesian model for signal synthesis, which is based on a Boltzmann machine, and designed specialized optimization methods for the estimation problems that arise from this model. This includes MAP estimation of the sparse representation and learning of the model parameters. The main contributions of this work include the exploration of settings where exact MAP estimation is possible and the development of an efficient message-passing algorithm for signal recovery in this setup. We also suggested a greedy algorithm for signal recovery which approximates the MAP estimator and uses the Bayesian framework to its full extent. After developing the recovery algorithms, we addressed learning issues and designed an efficient estimator for the parameters of the graphical model. Finally, we provided a comprehensive comparison between the suggested methods, along with standard sparse recovery algorithms. We demonstrated the effectiveness of our approach through synthetic experiments.

APPENDIX A
PROOF OF THEOREM 1

We show how the assumption that the dictionary is unitary can be used to simplify the expression for $\Pr(S|y)$. For a unitary dictionary we have $A_s^T A_s = I$ for any support s. Consequently, for a support of cardinality k the matrix $D = A_s^T A_s + \sigma_e^2 \Sigma_s^{-1}$ is a diagonal matrix of size k-by-k with entries $d_i = 1 + \sigma_e^2/\sigma_{x,i}^2$, $i = s_1, \ldots, s_k$ on its main diagonal. Straight forward computations show that the following relations hold:

$$\begin{aligned} y^T A_s D^{-1} A_s^T y &= \sum_{i \in s} d_i (y^T a_i)^2, \\ \ln\left((\det(D))\right) &= \sum_{i \in s} \ln\left(d_i\right) \end{aligned} \tag{37}$$

Using the definition of S ($S_i = 1$ implies that i is in the support and $S_i = -1$ implies otherwise), we can replace each sum over the entries in the support $\sum_{i \in s} v_i$ by a sum over all possible entries $\sum_{i=1}^m \frac{1}{2}\left(S_i + 1\right) v_i$. Consequently, the relations in (37) can be re-written as:

$$\begin{aligned} y^T A_s D^{-1} A_s^T y &= \frac{1}{2} \sum_{i=1}^m \left(S_i + 1\right) d_i (y^T a_i)^2 = C_1 + \frac{1}{2} f^T S \\ \ln\left((\det(D))\right) &= \frac{1}{2} \sum_{i=1}^m \left(S_i + 1\right) \ln\left(d_i\right) = C_2 + \frac{1}{2} g^T S \end{aligned} \tag{38}$$

where C_1, C_2 are constants and f, g are vector with entries $f_i = d_i (y^T a_i)^2$, $g_i = \ln\left(d_i\right)$ for $i = 1, \ldots, m$. Using the definition of D we place the relations of (38) into the appropriate terms in (10) and get:

$$\ln\left(\Pr(S|y)\right) = C_3 + \left(b + \frac{f}{4\sigma_e^2} - \frac{v}{4} - \frac{g}{4}\right)^T S + \frac{1}{2} S^T W S \tag{39}$$

where C_3 is a constant. It is now easy to verify that the posterior distribution $\Pr(S|y)$ corresponds to a BM distribution with the same interaction matrix W and a modified bias vector which we denote by $q = b + \frac{f}{4\sigma_e^2} - \frac{v}{4} - \frac{g}{4}$:

$$\Pr(S|y) = \frac{1}{\widetilde{Z}} \exp\left(q^T S + \frac{1}{2} S^T W S\right) \tag{40}$$

where $\widetilde{Z}$ is a partition function of the BM parameters W, q which normalizes the distribution. Using the definitions of f, g and v we get that (15) holds.

REFERENCES

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[2] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[3] C. La and M. N. Do, "Tree-based orthogonal matching pursuit algorithm for signal reconstruction," in *ICIP*, Atlanta, GA, Oct. 2006.

[4] Y. M. Lu and M. N. Do, "Sampling signals from a union of subspaces," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 41–47, Mar. 2008.

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

[6] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Compressed sensing of block-sparse signals: uncertainty relations and efficient recovery," *IEEE Trans. Signal Processing*, vol. 58, no. 6, pp. 3042–3054, June 2010.

[7] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, (to appear).

[8] M. Duarte and Y. C. Eldar, "Structured compressed sensing: from theory to applications," *IEEE Trans. Signal Processing*, (submitted).

[9] S. G. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[10] M. Mishali and Y. C. Eldar, "Reduce and boost: recovering arbitrary sets of jointly sparse vectors," *IEEE Trans. Signal Processing*, vol. 56, no. 10, pp. 4692–4702, Oct. 2008.

[11] M. Mishali and Y. C. Eldar, "Blind multi-band signal reconstruction: compressed sensing for analog signals," *IEEE Trans. Signal Processing*, vol. 57, no. 3, pp. 993–1009, Mar. 2009.

[12] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

[13] M. W. Seeger, "Bayesian inference and optimal design for the sparse linear model," *Journal of Machine Learning Research*, vol. 9, pp. 759–813, Apr. 2008.

[14] P. Schnitter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," in *ITA*, La Jolla, CA, Jan. 2008.

[15] J. S. Turek, I. Yavneh, M. Protter, and M. Elad, "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary," Tech. Rep., CS Dept., Technion – Israel Institite of Technology, Haifa, Israel, 2010.

[16] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, no. 4, pp. 211–244, 2001.

[17] R. M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Tech. Rep., CS Dept., University of Toronto, 2003.

[18] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[19] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4701–4714, Oct. 2009.

[20] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden Markov tree model," in *ICASSP*, Las Vegas, NV, Apr. 2008.

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

[22] 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 20*, J.C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., pp. 505–512. 2008.

[23] V. Cevher, M. F. Duarte, C. Hedge, and R. G. Baraniuk, "Sparse signal recovery using Markov random fields," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 257–264. 2009.

[24] M. I. Jordan, "Graphical models," *Statistical Science*, vol. 19, no. 1, pp. 140–155, 2004.

[25] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[26] A. Wiesel, Y. C. Eldar, and A. O. Hero III, "Covariance estimation in decomposable Gaussian graphical models," *IEEE Trans. Signal Processing*, vol. 58, no. 3, pp. 1482–1492, Mar. 2010.

[27] D. Z. Du and P. M. Pardalos, Eds., *Handbook of Combinatorial Optimization*, vol. 3, pp. 1–19, Kluwer Academic Publishers, 1998.

[28] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

[29] J. Besag, "Statistical analysis of non-lattice data," *The Statistican*, vol. 24, pp. 179–195, 1975.

[30] A. Hyvarinen, "Consistency of pseudolikelihood estimation of fully visible boltzmann machines," *Neural Computation*, vol. 18, no. 10, pp. 2283–2292, Oct. 2006.

[31] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[32] G. Narkiss and M. Zibulevsky, "Sequential subspace optimization method for large-scale unconstrained optimization," Tech. Rep., EE Dept., Technion – Israel Institite of Technology, Haifa, Israel, 2005.

[33] T. Faktor, Y. C. Eldar, and M. Elad, "Denoising of image patches via sparse representations with learned statistical dependencies," in *ICASSP*, Prague, Czech Republic, May 2011, (submitted).