MAP ESTIMATION FOR BAYESIAN MIXTURE MODELS WITH SUBMODULAR PRIORS

Marwa El Halabi, Luca Baldassarre and Volkan Cevher

LIONS, École Polytechnique Fédérale de Lausanne, Switzerland

ABSTRACT

We propose a Bayesian approach where the signal structure can be represented by a mixture model with a submodular prior. We consider an observation model that leads to Lipschitz functions. Due to its combinatorial nature, computing the maximum a posteriori estimate for this model is NP-Hard, nonetheless our converging majorization-minimization scheme yields approximate estimates that, in practice, outperform state-of-the-art methods.

Index Terms— Mixture models, Submodular, MAP estimate, Compressive sensing

1. INTRODUCTION

The problem of recovering a signal $\mathbf{x} \in \mathbb{R}^N$ from linear measurements $\mathbf{y} \in \mathbb{R}^M$ is ubiquitous, appearing in fields ranging from compressive sensing, linear regression and sparse linear models in machine learning. For instance, obtaining $\mathbf{x}$ from $\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$, where $\mathbf{A}$ is a $M \times N$ measurement matrix with $M < N$ and $\varepsilon \in \mathbb{R}^M$ is a random noise vector, is an ill-posed problem with infinitely many solutions. It is therefore necessary to have some prior structural information on the signal $\mathbf{x}$ in order to successfully recover it. In a Bayesian framework, this is done by placing a prior distribution on the signal that favors the desired structure.

For example, in compressive sensing, $\mathbf{x}$ is assumed to be sparse, that is only $K \ll N$ of its components are non-zero. This allows to circumvent the ill-posedeness of the problem and achieve guaranteed recovery using only $\mathcal{O}(K \log(K/N))$ samples [1, 2]. However, signals encountered in practice usually present more elaborate structures than simple sparsity. Exploiting this structure can help reduce the number of required samples, decrease recovery error, and allow better interpretability [3, 4].

We differentiate between two kinds of prior knowledge: a discrete structure on the state (e.g., zero/non-zero, or small/large) of the coefficients (e.g., the non-zero coefficients are grouped in given index sets [5]), and a continuous structure on the values of the coefficients of the signal (e.g., the coefficients are sampled from a Gaussian distribution with fixed variance). In this paper, we investigate models that leverage both types of structure.

We consider Bayesian mixture models where the signal is generated by a mixture of probability distributions. Mixture models provide flexibility to model real-world signals, and are often used as priors in practice (cf., Sect. 4.2). In particular, we assume each component x_i is independently drawn from one of two possible distributions $\mathcal{Q}_0$ and $\mathcal{Q}_1$, which corresponds to two possible states of x_i. To this end, we introduce for each x_i a latent binary random state variable $s_i \in \{0, 1\}$ which indicates the distribution x_i was drawn from, i.e., $x_i \sim \mathcal{Q}_{s_i}(\theta_{i,s_i})$ where θ_{i,s_i} are the parameters of $\mathcal{Q}_{s_i}$. The discrete structure is encoded by a prior distribution over the state vector $\mathbf{s}$ that ensures that certain state configurations are more likely than others. In particular, we assume that the discrete structure can be captured by a prior $p(\mathbf{s}) = \exp(-R(\mathcal{S}))$, where R is a *submodular* set function (cf., Definition 3) with parameters ψ and $\mathcal{S} = \{i | s_i = 1\}$ (we will use $\mathbf{s}$ and $\mathcal{S}$ interchangeably). Submodular set functions appear widely in applications, see for example [6, 7, 8, 9].

For simplicity, we assume all hyperparameters in our model $(\theta_{i,1}, \theta_{i,0}, \psi)$ are known. Learning the hyperparameters is deferred to future work. A graphical summary of the considered model is depicted in Figure 1, for the case where the noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ (cf., Sect. 4.1).

We propose to estimate $\mathbf{x}$ by computing its *maximum a posteriori* (MAP) estimate $\hat{\mathbf{x}}$. However, the presence of the discrete component $R(\mathcal{S})$ in our model renders the optimization difficult. We present an extension of the efficient Majorization-Minimization algorithm introduced in [10] that iteratively maximizes the log-posterior $\log p(\mathbf{x}, \mathbf{s}|\mathbf{y})$, with guaranteed convergence. Our numerical results show that the proposed algorithm can take full advantage of all available prior information on the signal, while for non-truly sparse signals, state-of-the-art methods are capable of leveraging only a part of it. For sparse signals, our algorithm can be used to further improve on convex methods.

2. NOTATION AND PRELIMINARIES

We denote scalars by lowercase letters, vectors by lowercase boldface letters, matrices by boldface uppercase letters, and sets by uppercase script letters. We represent

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, SNF 200021- 132548, SNF 200021-146750 and SNF CRSII2-147633.

978-1-4799-3694-6/14/$31.00

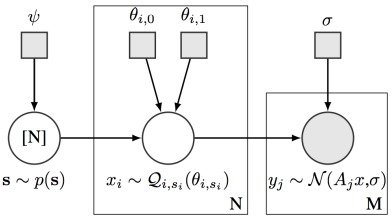

Fig. 1: *Graphical model*

the ground set of N indices by $\mathcal{N} = \{1, \ldots, N\}$. The i-th entry of a vector $\mathbf{x}$ is x_i. We now introduce some definitions that will be used in the following.

Definition 1. *A function $f : \mathbb{R}^N \to \mathbb{R}$ is a smooth Lipschitz continuous gradient function if $\forall\, \mathbf{x}, \mathbf{x}' \in dom(f)$, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\|_2 \leq L\|\mathbf{x} - \mathbf{x}'\|_2$, for some global constant $L > 0$.*

Definition 2. *We define the proximity operator of a function $g : \mathbb{R}^N \to \mathbb{R}$, as $prox_{\lambda g}(\mathbf{z}) := \arg\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 + \lambda g(\mathbf{x})$, where $\lambda > 0$ is a regularization parameter.*

Definition 3. *A set function $R : 2^{\mathcal{N}} \to \mathbb{R}$ is submodular iff it satisfies the following diminishing returns property: $\forall \mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{N}, \forall e \in \mathcal{N} \setminus \mathcal{T}$, $R(\mathcal{S} \cup \{e\}) - R(\mathcal{S}) \geq R(\mathcal{T} \cup \{e\}) - R(\mathcal{T})$. If this inequality is satisfied with equality everywhere, the function R is said to be* modular.

Submodularity is considered the discrete equivalent of convexity in the sense that submodular function minimization (SFM) admits efficient algorithms, with best known complexity of $O(N^5T + N^6)$, where T is the function evaluation complexity [11]. In practice, however, the minimum-norm point algorithm is usually used, which commonly runs in $O(N^2)$, but has no known complexity [12]. Furthermore, for certain functions which are "graph representable" [13, 14], SFM is equivalent to the minimum s-t cut on an appropriate graph $G(\mathcal{V}, \mathcal{E})$, with time complexity[1] $\tilde{O}(|\mathcal{E}| \min\{|\mathcal{V}|^{2/3}, |\mathcal{E}|^{1/2}\})$ [15].

3. OPTIMIZATION

In what follows, we denote the likelihood distribution by $p(\mathbf{y}|\mathbf{x}) = \exp(-\mathcal{L}_y(\mathbf{x}))$, where $\mathcal{L}_y(\mathbf{x})$ is some suitable data fidelity term. In our model, we make the following assumptions:

- A1 The loss function $\mathcal{L}_y(\mathbf{x}) = -\log p(\mathbf{y}|\mathbf{x})$ is smooth with L-Lipschitz continuous gradient (cf., Def. 1).
- A2 The variables x_i are independent given s_i, i.e. $\log p(\mathbf{x}|\mathbf{s}) = \sum_{i=1}^N \log p(x_i|s_i)$.
- A3 The function $G(\mathbf{x}|\mathbf{s}) = -\sum_{i=1}^N \log p(x_i|s_i)$ has an easy to compute proximal operator (cf., Def. 2).
- A4 The regularizer on the state vector $R(\mathcal{S}) = -\log p(\mathbf{s})$ is submodular (cf., Def. 3).

[1] the notation $\tilde{O}(\cdot)$ ignores log terms

We want to compute the MAP estimate of $[\mathbf{x}, \mathbf{s}]$.

$$\begin{aligned}[\widehat{\mathbf{x}}, \widehat{\mathbf{s}}] &= \arg\max_{\mathbf{x},\mathbf{s}} p(\mathbf{x}, \mathbf{s}|\mathbf{y}) \\ &= \arg\min_{\mathbf{x},\mathbf{s}} -\log p(\mathbf{y}|\mathbf{x}) - \log p(\mathbf{x}|\mathbf{s}) - \log p(\mathbf{s}) \\ &= \arg\min_{\mathbf{x},\mathbf{s}} \mathcal{L}_y(\mathbf{x}) - \sum_{i=1}^N \log p(x_i|s_i) + R(\mathcal{S}) \quad (1)\end{aligned}$$

Unfortunately, computing the MAP estimate (1) is NP-Hard: for instance the NP-Hard problem of minimizing the least square with ℓ_0 regularization [16] can be cast as a special case. Here, we aim to efficiently compute numerically good approximations to the MAP estimator.

Given our assumptions, the objective function in (1) can be iteratively minimized by the majorization-minimization scheme of Algorithm 1, see also [10]. The main idea is to majorize the continuous part $\mathcal{L}_y(\mathbf{x}) + G(\mathbf{x}|\mathbf{s})$ at each iteration by a modular upper bound, and then solve the resulting SFM. By assumption A1, the loss function admits the following quadratic upper bound:

$$\begin{aligned}\mathcal{L}_y(\mathbf{x}) &\leq \mathcal{L}_y(\mathbf{x}') + \langle \nabla \mathcal{L}_y(\mathbf{x}'), \mathbf{x} - \mathbf{x}' \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}'\|_2^2 \\ &= C(\mathbf{x}') + \frac{L}{2}\|\mathbf{x} - (\mathbf{x}' - \frac{1}{L}\nabla \mathcal{L}_y(\mathbf{x}'))\|_2^2 \\ &:= Q(\mathbf{x}, \mathbf{x}') \quad (2)\end{aligned}$$

$\forall\, \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N$, and where $C(\mathbf{x}')$ depends only on $\mathbf{x}'$.

Therefore, the objective function in (1) is upper bounded by $Q(\mathbf{x}, \mathbf{x}') + G(\mathbf{x}|\mathbf{s}) + R(\mathcal{S})$. At each iteration $j+1$, we minimize this upper bound with $\mathbf{x}' = \mathbf{x}^j$, the estimate obtained at the previous iteration,

$$\min_{\mathbf{x},\mathbf{s}} Q(\mathbf{x}, \mathbf{x}^j) + G(\mathbf{x}|\mathbf{s}) + R(\mathcal{S}) = \quad (3)$$

$$\min_{\mathbf{s}} \min_{\mathbf{x}} \frac{L}{2}\|\mathbf{x} - (\mathbf{x}^j - \frac{1}{L}\nabla \mathcal{L}_y(\mathbf{x}^j))\|_2^2 + G(\mathbf{x}|\mathbf{s}) + R(\mathcal{S})$$

Fixing the support $\mathbf{s}$, the minimization with respect to $\mathbf{x}$ reduces to a simple proximity operation, which by assumption A2 is easy to compute. Let $\widehat{\mathbf{x}}_{\mathbf{s}}^j = \mathrm{prox}_{G(\cdot|\mathbf{s})/L}(\mathbf{x}^j - \frac{1}{L}\nabla \mathcal{L}_y(\mathbf{x}^j))$ and define $M(\mathcal{S}) :=$

$$\sum_{i=1}^N \left(\frac{L}{2}\left(\widehat{x}_{i,\mathbf{s}}^j - (x_i^j - \frac{1}{L}\nabla_i \mathcal{L}_y(\mathbf{x}^j))\right)^2 - \log p(\widehat{x}_{i,\mathbf{s}}^j | s_i) \right)$$

Then the minimization in (3) is equivalent to:

$$\min_{\mathcal{S}} M(\mathcal{S}) + R(\mathcal{S}) \quad (4)$$

Since $M(\mathcal{S})$ is modular, (4) is a SFM that can be solved efficiently (cf., Sect. 2). Given the optimal state vector s^{j+1}, we update our estimate $\mathbf{x}^{j+1}$ by minimizing the original objective function with $\mathbf{s} = s^{j+1}$, if it can be done efficiently, otherwise we use $\mathbf{x}^{j+1} = \widehat{\mathbf{x}}_{s^{j+1}}^j$.

Proposition 1 (Convergence)**.** *Algorithm 1 produces a sequence $\mathbf{x}^{j+1}$ that satisfies $p(\mathbf{x}^{j+1}, \mathbf{s}^{j+1}|\mathbf{y}) \geq p(\mathbf{x}^j, \mathbf{s}^j|\mathbf{y})$ which implies convergence in the objective value.*

The proof of Proposition 1 follows from similar arguments as in [10].

Algorithm 1 MAP-MM algorithm

Input: $\mathbf{x}^0 \in \mathbb{R}^N$
while *not converged* **do**
$\widehat{\mathbf{x}}^j_{\mathbf{s}} = \text{prox}_{G(\cdot|\mathbf{s})/L}(\mathbf{x}^j - \frac{1}{L}\nabla\mathcal{L}_y(\mathbf{x}^j))$
$\mathbf{s}^{j+1} = \arg\min_{\mathbf{s}} Q(\widehat{\mathbf{x}}^j_{\mathbf{s}}, \mathbf{x}^j) + G(\widehat{\mathbf{x}}^j_{\mathbf{s}}|\mathbf{s}) + R(\mathcal{S})$
$\mathbf{x}^{j+1} = \arg\min_{\mathbf{x}} \mathcal{L}_y(\mathbf{x}) + G(\mathbf{x}|\mathbf{s}^{j+1})$
end while

4. MODELS

In this section, we present some examples of signal priors that fit in our framework.

4.1. Priors on the noise

By assumption A1, we consider noise priors that lead to Lipchitz continuous loss functions. For example when ε is a zero-mean Gaussian noise with covariance $\sigma^2 I$, i.e., $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2 I)$, the data fidelity term is the usual least squares loss function $\mathcal{L}_y(\mathbf{x}) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + M\log(\sqrt{2\pi}\sigma)$. Another example is the logistic loss function, commonly used in classification, $\mathcal{L}_y(\mathbf{x}) = \sum_{i=1}^M \log(1 + \exp(-y_i(\mathbf{a}_i^T\mathbf{x})))$, where $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$, which corresponds to the prior $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^M \frac{1}{1+\exp(-y_i(\mathbf{a}_i^T\mathbf{x}))}$.

4.2. Priors on the continuous structure of signal

We consider each coefficient x_i to be the mixture of two distributions that results in a separable function $G(\mathbf{x}|\mathbf{s})$ with an easy to compute proximity operator (cf., A2 and A3). The Gaussian and Laplacian distributions are examples of distributions that can be used, since both have closed form proximity operators. For $p(x_i|s_i) = \mathcal{N}(\mu_{i,s_i}, \sigma^2_{i,s_i})$, the proximity operation used in Algorithm 1 reduces to:

$$\widehat{x}^{j+1}_{i,\mathbf{s}} = \frac{L(x_i^j - \frac{1}{L}\nabla_i\mathcal{L}_y(\mathbf{x}^j)) + \mu_{i,s_i}/\sigma^2_{i,s_i}}{L + 1/\sigma^2_{i,si}}$$

And for $p(x_i|s_i) = \text{Laplace}(\mu_{i,s_i}, \sigma_{i,s_i})$, it becomes:

$$\widehat{x}^{j+1}_{i,s_i} = \mu_{i,s_i} + \text{Soft}\left(x^j - \frac{1}{L}\nabla_i\mathcal{L}_y(\mathbf{x}^j) - \mu_{i,s_i}, 1/(L\sigma_{i,s_i})\right)$$

where $\text{Soft}(x,\tau) = \max(|x| - \tau, 0)\text{sign}(x)$ is the standard soft-thresholding operator.
The mixture of two (or more) Gaussians, i.e $\mathcal{Q}_{s_i}(\theta_{i,s_i}) = \mathcal{N}(\mu_{i,s_i}, \sigma^2_{i,s_i})$ such that $\sigma_{i,1} > \sigma_{i,0}$, is ubiquitous in literature (See for e.g., [17, 18, 19]), due to their simplicity and effectiveness in modeling real-world signals. One can also use a Gaussian-Laplacian mixture, i.e $\mathcal{Q}_{s_1}(\theta_{i,s_1}) = \mathcal{N}(\mu_{i,1}, \sigma^2_{i,1})$, and $\mathcal{Q}_{s_0}(\theta_{i,s_0}) = \text{Laplace}(\mu_{i,0}, \sigma_{i,0})$ where the Laplacian distribution is used as sparsity promoting prior [20]. Another example is the laplacian mixture model, an analogue to the Gaussian mixture model, that is better suited to model signals with "peaky" distributions (See for e.g., [21, 22, 23]).

4.3. Priors on the discrete structure of signal

We consider priors on the hidden binary variables $\mathbf{s}$ that yield a submodular function $R(\mathcal{S})$, with $\mathcal{S} = \{i|s_i = 1\}$ (cf., A4). We provide below 3 examples of discrete structures, encountered in practice, that satisfy this assumption. In what follows, we refer to coefficients with $s_i = 1$ ($s_i = 0$) as "large" ("small") coefficients, since this state is associated with the distribution of larger (smaller) variance $\sigma_{i,1}$ ($\sigma_{i,0}$) (cf., Sect. 4.2).

4.3.1. Approximately sparse model

The simplest discrete prior on $\mathbf{x}$ is the expected number K of large coefficients, which is the sparsity for signals whose "small" coefficients are exactly zero. In this model, each binary variable s_i is drawn independently from the same Bernoulli distribution with known parameter K/N. We then have $p(\mathbf{s}) = \prod_{i=1}^N (\frac{K}{N})^{s_i}(1-\frac{K}{N})^{1-s_i}$ and

$$-\log p(\mathbf{s}) = \sum_{i=1}^N \left(s_i \log\left(\frac{N-K}{K}\right) - \log\left(1 - \frac{K}{N}\right)\right)$$

which is a modular function over the indicator variable $\mathbf{s}$.

When $K \ll N$, this discrete prior used in conjunction with the mixture model (cf., Sect. 4.2) captures well the structure of approximately sparse signals, where "small" values are not small enough to be ignored. Note that for $\sigma_0 = 0$, we recover the standard sparsity model.

Sparse Gaussian mixtures were also considered in [24] for compressive sensing with approximately sparse signals, but the proposed method relies on a particular measurement scheme, while our approach assumes that the measurement matrix is general.

4.3.2. Markov Tree model

Moving beyond simple sparsity priors, one can consider priors where each binary variable s_i is drawn from a Bernoulli distribution with parameters that depends on the index i. In particular, we consider the Markov Tree Gaussian mixture model described in [17] which assumes that the variables x_i are organized over a given tree, and their values tend to decay from root to leaves. This model provides a good description of wavelet coefficients encountered in many classes of signals [17].

Formally, we have $p(\mathbf{s}) = \prod_{i=1}^N \mathcal{B}(1, p_i)$, where p_i depend on the level of the variable x_i in the tree, so that

$$-\log p(\mathbf{s}) = \sum_{i=1}^N \left(s_i \log\left(\frac{1-p_i}{p_i}\right) - \log(1-p_i)\right)$$

which is again modular.

4.3.3. Ising model

The Ising model [25] is used to capture the clustering of "large" coefficients in a signal, a desired structure for example in background subtraction in images or videos [7].

The signal structure is encoded on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices are the indices $\mathcal{V} = \mathcal{N}$ and the edges connect neighboring coefficients. For example, for images, the vertices are the pixels of the image and edges connect pixels next to each other, forming the so-called two dimensional lattice Ising model. The Ising penalty is then expressed via the following symmetric submodular function:

$$R_{\text{ISING}}(\mathbf{s}) = \sum_{(i,j)\in\mathcal{E}} \iota(s_i \neq s_j) \tag{5}$$

where ι is an indicator function such that $\iota(s_i \neq s_j) = 1$ if $s_i \neq s_j, 0$ otherwise. A clustered sparse signal can be modelled by the following prior:

$$p(\mathbf{s}) \propto \exp(-\lambda R_{\text{ISING}}(\mathcal{S}) - \rho|\mathcal{S}|)$$

for certain parameters $\lambda, \rho \geq 0$ that control the level of sparsity and "clusteredness".

Remark 1. *Note that, since the Approximately sparse model and Markov Tree model yield modular regularizers, our algorithm can easily handle more than* 2 *states with these priors.*

5. SIMULATIONS

We demonstrate our approach on the two state Gaussian mixture model in conjunction with the 3 discrete priors described in Section 4.3. An example of a signal generated by each model is shown in Figure 2.

We consider a linear model, $\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ and $\mathbf{A}$ a random normalized Gaussian matrix. We measure the relative recovery error with $E = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$ and the state variables recovery quality with $Q = \|\hat{\mathbf{s}} - \mathbf{s}\|_0$. We fix $\sigma = 0.01$, $\sigma_0 = 1$ and $\sigma_1 = r$, with $r = 10$ and $r = 100$. The value of r controls the sparsity of the signal; a small r leads to signals not truly sparse (cf., Fig.2), while a large r leads to sparser signals. We adopt two initializations for our proposed algorithm, MAP-MM starts with $\mathbf{x}^0 = \mathbf{0}$, while MAP-MM-I starts with the estimate of the best convex competing method. Figures 3, 4, and 5 (left) illustrate the importance of a correct initialization of MAP-MM in the sparse case $r = 100$, where convex approaches capture well the structure of the signal: starting from their estimate allows MAP-MM to achieve further improvement. For the non sparse case $r = 10$, even MAP-MM obtains excellent performance, as shown in Figures 3, 4, and 5 (middle).

We fix the dimension $N = 1024$, and vary M from 128 to 1024 measurements. For each M we perform 50 simulations using different randomly generated signals and measurement matrices.

For recovery of the state variables, we only consider MAP-MM and Orthogonal Matching Pursuit (OMP) [26], since all the other algorithms considered cannot recover the state variables. As a baseline, we compare against the recovery quality Q achieved by always guessing small (red dashed line). MAP-MM (or MAP-MM-I for $r = 100$) always outperforms OMP in terms of recovering the correct states (Figs. 3, 4, and 5 (right))

5.1. Approximately Sparse Gaussian mixture model

We consider signals where each s_i is sampled from $\mathcal{B}(0, \frac{K}{N})$ with $K = 128$, then each x_i is independently drawn from $\mathcal{N}(0, \sigma^2_{s_i})$ with the same large/small variances for all $i \in \mathcal{N}$. Figure 3 shows the performance of MAP-MM, OMP and Basis pursuit denoising (BPDN) [27], as we vary M. OMP only exploits the true number of large coefficients in $\mathbf{x}$ which is clearly not enough for signals with non-negligible small coefficients (middle). BPDN uses the true variance of the noise and also accounts for sparsity by way of the ℓ_1-norm, yielding better estimated than OMP, but still worse than MAP-MM.

5.2. Hidden Markov Tree Gaussian mixture model

We consider the Markov Tree Model proposed in [17] using a binary tree. We assume that the root is always picked as a "large" coefficient with variance σ_1^2, while its child is either large with probability $p_i = 0.9$ and variance σ_1^2, or small with probability 0.1 and variance σ_0^2. The large and small variances decay according to the level and p_i depends both on the level and the state of its parent. For details, we refer to [17]. We use the following parameters: $\alpha_0 = 0.2$, $\alpha_1 = 0.1$, $C_{11} = 0.5$, $C_{00} = 2$, $\gamma_0 = 5$ and $\gamma_1 = 0.5$. This choice implies that the coefficients states are persistent across levels and the variances decay slowly, so that small coefficients at deep levels are still not negligible.

Figure 4 shows the performance of MAP-MM, OMP, BPDN, Hierarchical group lasso (HGL) [28] and the weighted BP algorithm with weights defined as the probability of being large (WBPDN) [17], as we vary M. MAP-MM-I outperforms all the other algorithms for both $r = 10$ and $r = 100$. Even though WBPDN and HGL leverage the tree structure, they do not take into account the coefficients variances and hence produce poorer results.

5.3. Sparse Ising Gaussian mixture model

We consider the one dimensional Ising model over a chain. We sample $\mathbf{s}$ from $p(\mathbf{s}) \propto \exp(-\lambda R_{\text{ISING}}(\mathcal{S}) - \rho|\mathcal{S}|)$ with the parameters $\lambda = 5$ and $\rho = 1.0986$ that yield an average sparsity of 113. Each $\mathbf{x}_i$ is then independently drawn from $\mathcal{N}(0, \sigma^2_{s_i})$, with the same large/small variances for all $i \in \mathcal{N}$.

Figure 5 shows the performance of MAP-MM, OMP, BPDN, overlapping group lasso (OGL) [5] with sequential groups of length 2 and overlap 1 and Fused Lasso

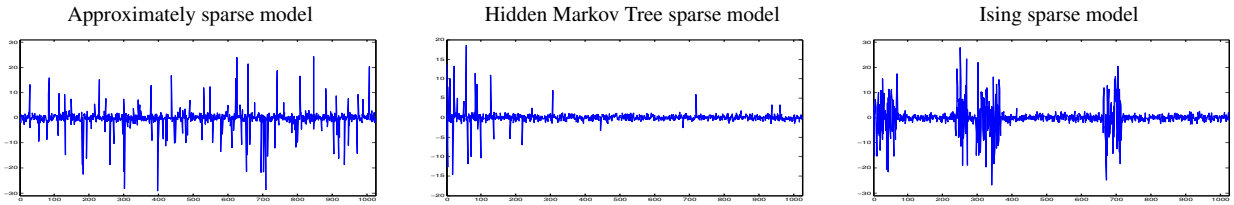

Fig. 2: *Signals sampled from each model, for $\sigma_1/\sigma_0 = 10$ and other parameters as described in the text.*

(FLasso) [29], as we vary M. MAP-MM-I again outperforms all the other algorithms. Both OGL and FLasso promote a clustering and sparsification of the coefficients, but do not exploit the continuous prior, yielding suboptimal performance.

6. CONCLUSIONS

We proposed a Bayesian approach for recovering structured signals generated by mixtures models with submodular priors and a majorization-minimization iterative scheme for obtaining the corresponding MAP estimate. In contrast to convex methods, our mixed convex-discrete criterion can exploit all available prior information on the structure of the signals and improve on the convex estimates. We are currently investigating theoretical characterizations of the best achievable performance of this approach.

7. REFERENCES

[1] E. J. Candes, "Compressive sampling," in *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[2] D.L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde, "Model-based compressive sensing," *Information Theory, IEEE Transactions on*, vol. 56, no. 4, pp. 1982–2001, 2010.

[4] L. Baldassarre, N. Bhan, V. Cevher, and A. Kyrillidis, "Group-sparse model selection: Hardness and relaxations," *arXiv preprint arXiv:1303.3207*, 2013.

[5] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, 2011.

[6] F. Bach, "Learning with submodular functions: A convex optimization perspective," *arXiv preprint arXiv:1111.6453*, 2011.

[7] V. Cevher, C. Hegde, M.F. Duarte, and R.G. Baraniuk, "Sparse signal recovery using markov random fields," in *NIPS*, 2009.

[8] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, 2004.

[9] M. Seeger, "On the submodularity of linear experimental design," Tech. Rep., 2009.

[10] M. El Halabi, L. Baldassarre, and V. Cevher, "To convexify or not? regression with clustering penalties on graphs," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*. IEEE, 2013.

[11] J.B. Orlin, "A faster strongly polynomial time algorithm for submodular function minimization," *Mathematical Programming*, vol. 118, no. 2, 2009.

[12] S. Fujishige and S. Isotani, "A submodular function minimization algorithm based on the minimum-norm base," *Pacific Journal of Optimization*, vol. 7, no. 1, pp. 3–17, 2011.

[13] S. Jegelka, H. Lin, and J. Bilmes, "On fast approximate submodular minimization.," in *NIPS*, 2011, pp. 460–468.

[14] S. Fujishige and S. Patkar, "Realization of set functions as cut functions of graphs and hypergraphs," *Discrete Mathematics*, vol. 226, no. 1, pp. 199–210, 2001.

[15] A. Goldberg and S. Rao, "Beyond the flow decomposition barrier," *J. ACM*, vol. 45, no. 5, pp. 783–797, Sept. 1998.

[16] Xiaoming Huo and Xuelei Ni, "When do stepwise algorithms meet subset selection criteria?," *The Annals of Statistics*, pp. 870–887, 2007.

[17] M.F. Duarte, M.B. Wakin, and R.G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden markov tree model," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008.

[18] S.D. Babacan, S. Nakajima, and M.N. Do, "Bayesian group-sparse modeling and variational inference," *Submitted to IEEE Transactions on Signal Processing*, 2012.

[19] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, 2008.

[20] M.W. Seeger, "Bayesian inference and optimal design for the sparse linear model," *The Journal of Machine Learning Research*, vol. 9, 2008.

[21] P. Garrigues and B.A. Olshausen, "Group sparse coding with a laplacian scale mixture prior," in *NIPS*, 2010.

[22] T. Amin, M. Zeytinoglu, and L. Guan, "Application of laplacian mixture model to image and video retrieval," *Multimedia, IEEE Transactions on*, vol. 9, no. 7, 2007.

[23] N. Mitianoudis and T. Stathaki, "Overcomplete source separation using laplacian mixture models," *IEEE Signal Processing Letters*, vol. 12, no. 4.

[24] S. Sarvotham, D. Baron, and R.G. Baraniuk, "Compressed sensing reconstruction via belief propagation," *preprint*, 2006.

[25] B. McCoy and T. Wu, *The Two-Dimensional Ising Model*, Harvard University Press, 1973.

[26] Joel A Tropp and Anna C Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[27] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[28] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[29] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

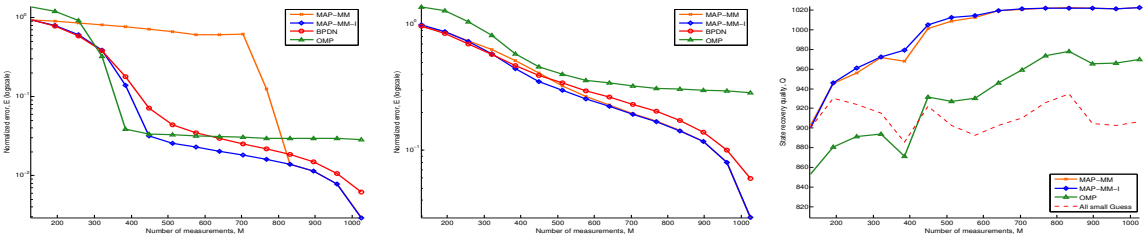

Fig. 3: *Performance of MAP-MM compared to other state-of-the arts algorithms for the approximately sparse Gaussian mixture model, in terms of signal recovery error E for $\sigma = 0.01, r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 129.*

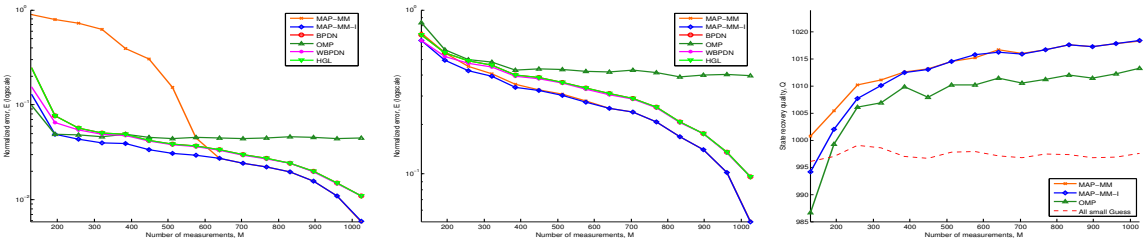

Fig. 4: *Performance of MAP-MM compared to other state-of-the arts algorithms for the Hidden Markov Tree Gaussian mixture model, in terms of signal recovery error E for $\sigma = 0.01, r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 27.*

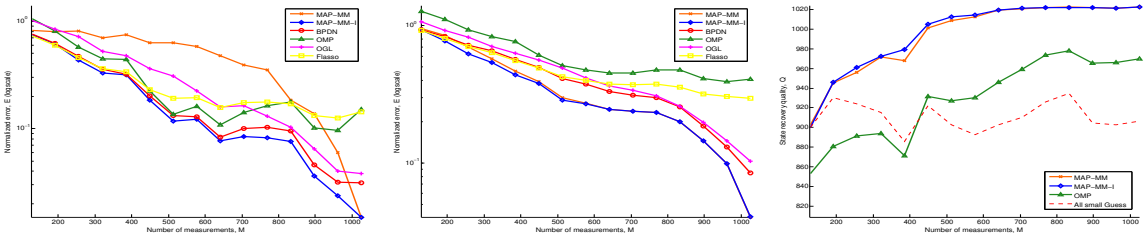

Fig. 5: *Performance of MAP-MM compared to other state-of-the arts algorithms for the sparse Ising Gaussian mixture model, in terms of signal recovery error E for $\sigma = 0.01, r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 113.*