Bayesian Compressive Sensing for Cluster Structured Sparse Signals

Lei Yu, Hong Sun, Jean-Pierre Barbot, Gang Zheng

Bayesian Compressive Sensing for Cluster Structured Sparse Signals

L. Yu[a,b,*], H. Sun[a], J. P. Barbot[b,c], G. Zheng[c]

[a] *E.I.S, Wuhan University, 129 Road of Luoyu, 430079 Wuhan, China*
[b] *ECS-Lab ENSEA, 6 Avenue du Ponceau, 95014 Cergy-Pontoise, France*
[c] *Group Non-A, INRIA, 59000 Lille, France*

Abstract

In traditional framework of Compressive Sensing (CS), only sparse prior on the property of signals in time or frequency domain is adopted to guarantee the exact inverse recovery. Other than sparse prior, structures on the sparse pattern of the signal have also been used as an additional prior, called *model-based compressive sensing*, such as clustered structure and tree structure on wavelet coefficients. In this paper, the cluster structured sparse signals are investigated. Under the framework of *Bayesian Compressive Sensing*, a hierarchical Bayesian model is employed to model both the *sparse prior* and *cluster prior*, then Markov Chain Monte Carlo (MCMC) sampling is implemented for the inference. Unlike the state-of-the-art algorithms which are also taking into account the *cluster prior*, the proposed algorithm solves the inverse problem automatically - prior information on the number of clusters and the size of each cluster is unknown. The experimental results show that the proposed algorithm outperforms many state-of-the-art algorithms.

[*] Corresponding author

Email addresses: yuleiwhu@gmail.com (L. Yu), hongsun@whu.edu.cn (H. Sun), barbot@ensea.fr (J. P. Barbot), gang.zheng@inria.fr (G. Zheng)

Keywords: Compressive Sensing, Cluster Structured Sparse Signals, Hierarchical Bayesian Model, MCMC.

1. Introduction

Compressive Sensing (CS) provides an alternative to Shannon/Nyquist sampling when signal under acquisition is known to be sparse or compressible [1, 2, 3, 4]. In the framework of CS, signals are measured through inner products with random vectors, and thus fewer measurements than periodic samples are needed: for any N dimensional signal x, its measurements v are taken as follows:

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon} = \boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{\theta} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{A} \in \mathcal{R}^{M \times N}$ is the sensing matrix, $\boldsymbol{\Psi} \in \mathcal{R}^{N \times N}$ is the sparse representation matrix with $\boldsymbol{\theta}$ the sparse coefficients and $\boldsymbol{\epsilon}$ is the noise item comprised of possible measurement noise and sparse representation errors. Without loss of generality, we denote the matrix multiplication $\boldsymbol{A\Psi}$ a single matrix[1] $\boldsymbol{\Phi}$, then (1) could be rewritten as:

$$\boldsymbol{v} = \boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\epsilon} \tag{2}$$

where the matrix $\boldsymbol{\Phi}$ is rank deficient with $M < N$, and hence loses information in general. However, it can be shown to preserve the information in sparse or compressible signals if it satisfies the so-called *Restricted Isometry Property* (RIP) [5]. To inverse the process (2), i.e., reconstruction of the

[1] Hereafter, $\boldsymbol{\Phi}$ is called sensing matrix instead of $\boldsymbol{A}$.

original sparse signal, a sparse promoting scheme is often exploited, such as

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \mathcal{R}^N} \frac{1}{2} \|\boldsymbol{v} - \boldsymbol{\Phi\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_p \tag{3}$$

where $\|\cdot\|_p$ represents the ℓ_p norm with $p \in [0, 1]$, and if $p = 0$ it corresponds to IHT [6] algorithm, if $p \in (0, 1)$ it corresponds to the Iterative Reweighted algorithm [7], if $p = 1$ it corresponds to the typical formula of LASSO (also for BPDN, IST [8] ...) problem. Moreover, the parameter λ is to balance the observation fitness and the sparse prior.

Besides the sparse property of nature signals (through sparse representation), the coefficients of sparse representation often exhibit as special structures, which can be exploited as the known information, and heuristically promote the performance of the reconstruction. Consider the N dimensional S-sparse signal $\boldsymbol{x}$ with Ω the set of locations of its nonzero entries, i.e. $\Omega = \mathrm{supp}(\boldsymbol{x})$, then one can define a subspace $\chi(\Omega) = \{\boldsymbol{x} : \boldsymbol{x}_\Omega \in \mathcal{R}^S, \boldsymbol{x}_{\bar{\Omega}} = 0\} \subset \mathcal{R}^N$ with $\boldsymbol{x}_\Omega$ the vector composed by the entries in Ω and $\boldsymbol{x}_{\bar{\Omega}}$ the vector composed by the entries not in Ω. Hence one can define a union of subspace $\mathcal{A} = \cup_{i=1}^{m_S} \chi(\Omega_i)$ with $m_S = \binom{N}{S}$ such that all S-sparse signals $\boldsymbol{x} \in \mathcal{A}$. Define $\delta_\mathcal{A}$ the constant of RIP for a sub-Gaussian sensing matrix $\Phi \in \mathcal{R}^{M \times N}$, if [9]

$$M \geq \frac{2}{c\delta_\mathcal{A}^2} \left(\ln(2m_S) + S \ln \frac{12}{\delta_\mathcal{A}} + t \right) \tag{4}$$

then its RIP is held for all elements in $\mathcal{A}$ with the probability $1 - e^{-t}$, in other words, the exact recovery is guaranteed with probability $1 - e^{-t}$. From [9], the bound for the number of measurements can be easily extended to the structured sparse signals with the same configuration except that the subspaces are limited to typical structures, and hence the number of subspaces will be largely decreased, i.e. $m_S \ll \binom{N}{S}$. In other words, the required number of measurements for structured sparse signals is much less than unstructured sparse signals.

The above analysis is heuristical and has been discussed in lots of literatures [9, 10, 11, 12, 13, 14]. Meanwhile, algorithms exploiting the structures as well as the sparsity have been exhaustively investigated in the aforementioned literatures. In this paper, we focus on *clustered sparsity model*[2], which is used in some applications where the significant coefficients of a sparse signal appear in clustered blocks. This kind of sparse pattern is often exploited in many concrete applications, such as multi-band signals, gene expression levels, source localization in sensor networks, MIMO channel equalization, magnetoencephalography [10, 9, 13].

The existing recovery algorithms for *clustered sparsity model* could be categorized into the following classes: 1) *Block Greedy Algorithms* [10, 13]; 2) *Dynamic Programming Method* [12]; 3) *Block Greedy with Statistical Model* [11]. To the best of the authors' knowledge, although all three classes of algorithms have taken into account the *cluster prior*, they also bring some new unknown parameters, such as the size and the number of the clusters, which, practically, are not easily obtained. For Class 1, Block-CoSaMP [10] and Block-OMP [13], the location of the clusters are fixed and the size (also the number) of clusters are required in the recovery procedure; for Class 2, the proposed algorithm in [12] does not require any information about the *cluster prior* except for the number of clusters; for Class 3, LaMP [11] models clustered sparse model with Markov Random Fields (MRF) and exploits

[2] It is also called block sparsity model in some other literatures.

the Matching Pursuit (MP) [15] procedure to carry out the sparse promoting. Additionally, it is worth noting that besides the *cluster prior*, sparsity information is necessary for all of the aforementioned algorithms.

In a probabilistic, Bayesian approach, through Graphical Models (GMs) [16], latent variables are often exploited to describe the dependencies (or joint probability distributions) between observations and parameters. It is usually called Latent Variable Analysis (LVA) [17], and possibly, results in some non-parametric approaches to Bayesian estimators. Exploiting sparsity probabilistic model [18], many algorithms based on the LVA are proposed to solve the sparse decomposition problems [19, 11, 20, 21, 22, 14]. Moreover, the structures of the sparse coefficients can be conveniently introduced into the LVA framework using Graphical Models [18, 11, 14]. Particularly, [18] has made a review on sparse signal recovery with GMs and introduced a GMs-based algorithm, exploiting cluster structure through the so-called Ising model [11], called LaMP. However, LaMP is actually following a greedy procedure constrained by the latent support variables, which are optimized through the graph cut algorithm at every iteration. Consequently, LaMP is not a systematic Bayesian approach.

In this paper, we employ a hierarchical Bayesian framework to model the *sparse prior* and the *cluster prior* simultaneously. The posterior distributions of the proposed prior model can be calculated. Nevertheless, no closed-form expressions of the Bayesian estimators can be derived and thus an MCMC-simulation scheme is required to implement the inference. It is different from any of the existing algorithms for *clustered sparsity model.* Since that the hierarchical Bayesian model allows the hyperparameters to be estimated in an unsupervised manner, the proposed algorithm does not require any information for both the sparse prior and the cluster prior. Moreover, unlike LaMP which exploits the MP procedure, the proposed algorithm is based on the Bayesian CS framework, where "deleting" process is also carried out during basis selecting iterations (MP does not have) and thus will not suffer worse case when selecting a wrong basis [21].

The paper is organized as below. In section 2, a hierarchical Bayesian generative model is proposed to take into account both the *sparse prior* and the *cluster prior*. In section 3, the posterior of the proposed Bayesian model is calculated and then an MCMC sampling is adopted for Bayesian inference. The experiments are carried out to illustrate the efficiency of the proposed recovery algorithm in section 4 and the paper ends up with a conclusion.

2. Bayesian CS for Cluster Structured Sparse Signals

2.1. Observation Likelihood

First, assume noises are white, i.e. obeys Gaussian distribution with zero mean and variance σ_0, and set $\alpha_0 = \sigma_0^{-1}$, then it has

$$\boldsymbol{v}|_{\boldsymbol{\theta},\alpha_0} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{\theta}, \alpha_0^{-1}\boldsymbol{I}) \tag{5}$$

where the notation $\boldsymbol{v}|_{\boldsymbol{\theta},\alpha_0}$ means that the random variable $\boldsymbol{v}$ depends on $\boldsymbol{\theta}$ and α_0.

After that, considering the conjugate prior to Gaussian distribution, a Gamma hyper prior is assigned on the hyperparameter α_0, which is $\alpha_0|_{c,d} \sim \text{Gamma}(c, d)$.

2.2. A Priori Model on Sparsity and Cluster

The unknown coefficients $\boldsymbol{\theta}$ are assigned a *prior* distribution $p(\boldsymbol{\theta})$, which models our knowledge on the nature of $\boldsymbol{\theta}$, i.e. *sparse* and *clustered.* In this subsection, both sparsity and cluster prior are simultaneously modeled through a "spike-and-slab" prior model, also called *Bernoulli-Gaussian process* [23, 24, 25], which has been widely used as a sparse promoting prior [26, 27, 28, 29].

2.2.1. Sparsity Prior

To model the sparseness of the coefficients $\boldsymbol{\theta}$, a "spike-and-slab" prior model is employed for each of the element θ_i, with π_i a mixing weight and δ_0 a point mass concentrated at zero.

$$\theta_i|_{\alpha_i,\pi_i} \sim (1-\pi_i)\delta_0 + \pi_i\mathcal{N}(0,\alpha_i^{-1}) \tag{6}$$

with α_i the precision (inverse-variance) of a Gaussian density function. Implicitly, the mixing weight π_i is the prior probability of a non-zero element, namely, the large mixing weight π_i corresponds to a nonzero entry with large probability, while the small π_i tends to generate a zero entry. Further, in order to obtain an explicit posterior density function, a conjugate prior for the parameters are defined: a Gamma prior is considered for variable α_i, i.e.

$$\alpha_i|_{a,b} \sim \text{Gamma}(a,b) \tag{7}$$

Implicitly, the pair of prior model (6) and (7) results in a sparse prior. Specifically, if $\pi_i = 0$, the only functional part is the point mass distribution concentrated at zero, hence all components of $\boldsymbol{\theta}$ equal zero. If $\pi_i = 1$, by marginalize over the hyperparameters $\boldsymbol{\alpha} \triangleq \{\alpha_i\}_{1:N}$, the overall prior on $\boldsymbol{\theta}$

with respect to a, b can be evaluated analytically through the integration over α_i,

$$p(\boldsymbol{\theta}|a, b) = \int p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|a, b)d\boldsymbol{\alpha} \propto \prod_{i=1}^{N} \left(b + \frac{\theta_i^2}{2}\right)^{-\left(a+\frac{1}{2}\right)}$$

which corresponds to the Student-t distribution [19]. And by setting $a, b \to 0$, the Student-t distribution can be reformulated as

$$p(\boldsymbol{\theta}) \propto \prod_{i=1}^{N} \frac{1}{|\theta_i|}$$

which is strongly peaked about $\theta_i = 0$. Consequently, the overall prior $p(\boldsymbol{\theta})$ favors sparseness.

2.2.2. Cluster Prior

To model the *cluster prior* of the coefficients $\boldsymbol{\theta}$, we must consider relations between the current element θ_i and its neighbors, called the *cluster pattern* of θ_i.

Definition 1 (k-th neighborhood). *Define the k-th neighborhood of location i over the coefficients $\boldsymbol{\theta}$, $\mathcal{U}_i^{(k)} = \{j|D(i, j) \leq k, j \neq i\}$ where $D(i, j)$ is the Euclidean distance between i and j with i and j the position on the vector $\boldsymbol{\theta}$.*

Denote $\mathcal{V}_N$ the set of all locations over the coefficients $\boldsymbol{\theta}$, i.e. $\mathcal{V}_N = \{1, ..., N\}$, then define $\mathcal{J}_{i,k,\otimes} \triangleq \mathcal{U}_i^{(k)} \cap \mathcal{V}_N$ and $\mathcal{J}_{i,k,\odot} \triangleq \mathcal{U}_i^{(k)} \cap \mathcal{V}_N \cup \{i\}$. Hence we can denote $\theta_{\mathcal{J}_{i,k,\otimes}}$ the set of components located in the neighbor of the ith coefficient θ_i, while denote $\theta_{\mathcal{J}_{i,k,\odot}}$ the set of components including both neighbors and the current component.

Then we categorize the relations into 3 different *cluster patterns* as shown in Fig. 1, where Pattern (a) denotes "all neighbors are zero" for θ_i, i.e. $\|\theta_{\mathcal{J}_{i,k,\otimes}}\|_0 = 0$, which corresponds to the isolated points, Pattern (b) denotes "part of neighbors are nonzero", i.e. $0 < \|\theta_{\mathcal{J}_{i,k,\otimes}}\|_0 < |\mathcal{J}_{i,k,\otimes}|$, which corresponds to the points located on the margin and Pattern (c) denotes "all neighbors are nonzero", i.e. $\|\theta_{\mathcal{J}_{i,k,\otimes}}\|_0 = |\mathcal{J}_{i,k,\otimes}|$, which corresponds to the clusters. In this place, $|\mathcal{J}|$ represents the cardinality of the set $\mathcal{J}$. Then according to the cluster patterns, the mixing weight π_i is chosen by the following *pattern selection procedure*:

$$\pi_i = \pi_i^{\langle q \rangle} \tag{8}$$

where $\pi_i^{\langle q \rangle}$ is drawn from different Beta distribution[3]: $\pi_i^{\langle q \rangle} \sim \text{Beta}(e^{\langle q \rangle}, f^{\langle q \rangle})$, with $q \in \{0, 1, 2\}$ corresponding respectively to pattern a, b and c.

In order to clarify the dependence within the random variables, the distributions for π could be rewritten as follows:

$$\pi_i|_{\boldsymbol{e},\boldsymbol{f},\theta_{\mathcal{J}_{i,k,\otimes}}} \sim p(\pi_i|\boldsymbol{e}, \boldsymbol{f}, \theta_{\mathcal{J}_{i,k,\otimes}}) \tag{9}$$

where $\boldsymbol{e} \triangleq \{e^{\langle q \rangle}\}_{q=0,1,2}$, $\boldsymbol{f} \triangleq \{f^{\langle q \rangle}\}_{q=0,1,2}$.

By considering the appropriate choice of parameters $\boldsymbol{e}, \boldsymbol{f}$, the cluster pattern selection procedure could promote the clusters and restrain the isolates. However, one may still be puzzled on the remained problems:

1). Neighborhood and cluster pattern: As shown in Fig. 1, only 1st neighborhood has been considered. Certainly, higher order neighborhood

[3] Since Beta distribution is a conjugate prior to Bernoulli likelihood with p the model parameters.

could be chosen, which, however, will result in lots of cluster patterns and thus make the pattern selection procedure more complicated. On the other hand, 1st neighborhood is enough, since relations between components and their neighbors can be spread around point by point.

2). Model parameters: The Beta(e, f) distribution tends to draw a small sample when $e < f$ and a big sample when $e > f$, while has no trends to only the big (or small) sample when $e = f$. Consequently, in order to promote clusters and restrain isolates, parameters $(e^{\langle 0\rangle}, f^{\langle 0\rangle})$, $(e^{\langle 1\rangle}, f^{\langle 1\rangle})$ and $(e^{\langle 2\rangle}, f^{\langle 2\rangle})$ must be chosen to drive the components with Pattern (a) to zero, which requests selecting a small π_i, and thus $e^{\langle 0\rangle} < f^{\langle 0\rangle}$. In opposite, $e^{\langle 2\rangle} > f^{\langle 2\rangle}$ for Pattern (c). While for Pattern (b), it can't be determined whether the current component tends to nonzero or not, and thus $e^{\langle 1\rangle} = f^{\langle 1\rangle}$. Further, empirically, the upper bound of these parameters must be small enough.

2.3. The Complete Generative Bayesian Model

Like the other generative model in Bayesian framework, the proposed model can be illustrated as well in a hierarchical structure, as shown in Fig. 2. Given the basic parameters on the top level, hyperparameters $\pi_i^{\langle 0\rangle}, \pi_i^{\langle 1\rangle}, \pi_i^{\langle 2\rangle}$ are drawn from Beta distribution with $\boldsymbol{e}$ and $\boldsymbol{f}$, then with the knowledge of cluster pattern of each components, the mixing weight π_i can be chosen by (8). Meanwhile, hyperparameters α_i are drawn from Gamma distribution with (a_0, b_0), and afterwards θ_i can be drawn through the "spike-and-slab" prior with π_i and α_i.

Unlike the model expressed in [21], the *cluster prior* is considered in the proposed model via the *pattern selection procedure*. Meanwhile, it is different from the Ising model expressed in [11], where Markov Random Field is considered and there is no explicit overall prior on sparse coefficients.

3. Bayesian Inference

In this section, we will adopt the Markov Chain Monte Carlo (MCMC) [30] to carry out the Bayesian inference of the proposed model. At first, denote the unknown parameters as $\mathcal{X} \triangleq \{\boldsymbol{\theta}, \alpha_0, \boldsymbol{\alpha}, \boldsymbol{\pi}\}$ and the model parameter as $\mathcal{M} \triangleq \{a, b, c, d, \boldsymbol{e}, \boldsymbol{f}\}$. Then the posterior distribution of $\mathcal{X}$ could be computed

$$p(\mathcal{X}|\boldsymbol{v}, \mathcal{M}) \propto p(\boldsymbol{v}|\mathcal{X})p(\mathcal{X}|\mathcal{M})$$

In addition, according to the hierarchical model described in Fig. 2, the conditional joint distribution $p(\mathcal{X}|\boldsymbol{v}, \mathcal{M})$ could be written as

$$p(\mathcal{X}|\boldsymbol{v}, \mathcal{M}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \alpha_0, \boldsymbol{\pi}, \boldsymbol{v})p(\boldsymbol{\alpha}|\boldsymbol{\theta}, a, b)p(\boldsymbol{\pi}|\boldsymbol{\theta}, \boldsymbol{e}, \boldsymbol{f})p(\alpha_0|\boldsymbol{\theta}, \boldsymbol{v}, c, d) \quad (10)$$

3.1. Posterior Distributions

3.1.1. Sparse signal $\boldsymbol{\theta}$

Assume that the components θ_i of the sparse signal $\boldsymbol{\theta}$ are *a priori* independent, namely, the full prior distribution of $\boldsymbol{\theta}$ can be rewritten as

$$p(\boldsymbol{\theta}|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \prod_{i}^{N} \left[(1 - \pi_i)\delta_0 + \pi_i \mathcal{N}(\theta_i|0, \alpha_i^{-1})\right]$$

Combining the observation likelihood $p(\boldsymbol{v}|\boldsymbol{\theta}, \alpha_0)$, one can compute the posterior distribution of $\boldsymbol{\theta}$ as follows

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{\pi}, \alpha_0, \boldsymbol{v}) &\propto p(\boldsymbol{\theta}|\boldsymbol{\pi}, \boldsymbol{\alpha})p(\boldsymbol{v}|\boldsymbol{\theta}, \alpha_0) \\ &= \left\{ \prod_{i}^{N} \left[(1 - \pi_i)\delta_0 + \pi_i \mathcal{N}(\theta_i|0, \alpha_i^{-1})\right] \right\} \mathcal{N}(\boldsymbol{v}|\boldsymbol{\Phi}\boldsymbol{\theta}, \alpha_0^{-1}\boldsymbol{I}) \end{aligned} \quad (11)$$

Denoting that $\mathbf{\Phi}_{-i}$ the sub matrix of $\mathbf{\Phi}$ excluding the ith column and $\boldsymbol{\theta}_{-i}$ the vector consisting of all but the ith component, one can design a Gibbs sampler for each component θ_i according to the following posterior distribution

$$p(\theta_i|\boldsymbol{\theta}_{-i}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \alpha_0, \boldsymbol{v}) \propto (1-\tilde{\pi}_i)\delta_0 + \tilde{\pi}_i \mathcal{N}(\theta_i|\tilde{\mu}_i, \tilde{\alpha}_i^{-1}) \tag{12}$$

with the parameters $\tilde{\pi}_i$, $\tilde{\mu}_i$ and $\tilde{\alpha}_i$ defined as follows

$$\begin{aligned}
\tilde{\alpha}_i &= \alpha_i + \alpha_0 \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i \\
\tilde{\mu}_i &= \tilde{\alpha}_i^{-1} \alpha_0 \boldsymbol{\phi}_i^T (\boldsymbol{v} - \mathbf{\Phi}_{-i}\boldsymbol{\theta}_{-i}) \\
\frac{\tilde{\pi}_i}{1-\tilde{\pi}_i} &= \frac{\pi_i}{1-\pi_i} \cdot \frac{\mathcal{N}(0|0, \alpha_i^{-1})}{\mathcal{N}(0|\tilde{\mu}_i, \tilde{\alpha}_i^{-1})}
\end{aligned}$$

3.1.2. Inverse variance $\boldsymbol{\alpha}$ of sparse model

Thanks to the conjugacy, the Gamma distribution on the inverse variance $\boldsymbol{\alpha}$ of sparse model leads to a straightforward posterior distribution for each element of $\boldsymbol{\alpha}$, written as

$$p(\alpha_i|\theta_{\mathcal{J}_{i,k,\odot}}) = \text{Gamma}(a + \frac{\|\theta_{\mathcal{J}_{i,k,\odot}}\|_0}{2}, b + \frac{\|\theta_{\mathcal{J}_{i,k,\odot}}\|_2^2}{2}) \tag{13}$$

3.1.3. Mixing weight $\boldsymbol{\pi}$

As depicted in Section 2, each element π_i of the mixing weight $\boldsymbol{\pi}$ is generated by selecting from three different parameters according to its corresponding sparsity pattern, i.e. for sparsity pattern $q \in \{0, 1, 2\}$, select $\pi_i = \pi_i^{\langle q \rangle}$.

On the other hand, for each sparsity pattern, the hyperparameter $\pi_i^{\langle q \rangle}$ obeys the Beta prior, which is conjugate to the Bernoulli distribution. Thus for sparsity pattern q, the posterior distribution of $\pi_i^{\langle q \rangle}$ can be easily calculated

$$p(\pi_i^{\langle q \rangle}|\theta_{\mathcal{J}_{i,k,\odot}}) = \text{Beta}(e^{\langle q \rangle} + \|\theta_{\mathcal{J}_{i,k,\odot}}\|_0, f^{\langle q \rangle} + |\mathcal{J}_{i,k,\odot}| - \|\theta_{\mathcal{J}_{i,k,\odot}}\|_0) \tag{14}$$

3.1.4. Noise variance α_0^{-1}

Similarly, α_0 is with Gamma distribution which is conjugate to the Gaussian distribution and thus one can easily compute the posterior distribution for α_0, written as

$$p(\alpha_0|\boldsymbol{\theta}, \boldsymbol{v}) = \text{Gamma}(c + \frac{M}{2}, d + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2) \tag{15}$$

3.2. Gibbs Sampler and MAP Estimation

Then, one can easily exploit the standard Gibbs sampler to generate the samples, and at each iterations, the detailed sampling steps can be expressed as Algorithm 1.

Algorithm 1 Standard Gibbs Sampler

```
For $i = 1, ..., N$
    sample $\theta_i$ from $p(\theta_i|\boldsymbol{\theta}_{-i}, \boldsymbol{\alpha}, \alpha_0, \boldsymbol{\pi}, \boldsymbol{v})$;
    sample $\alpha_i$ from $p(\alpha_i|\boldsymbol{\theta}, a, b)$;
    sample $\pi_i$ from $p(\pi_i|\boldsymbol{\theta}, \boldsymbol{e}, \boldsymbol{f})$;
End
sample $\alpha_0$ from $p(\alpha_0|\boldsymbol{\theta}, \boldsymbol{v}, c, d)$.
```

The purpose of the proposed Gibbs sampler in Algorithm 1 is to carry out the Bayesian inference of $\boldsymbol{\theta}$ and some other auxiliary hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\pi}$ and α_0, namely, $\mathcal{X}$. Through this Gibbs sampler, it will generate a set of collection of $\mathcal{X}$ asymptotically distributed according to the joint posterior of (10). Denote the set of this collection as follows.

$$\mathbf{X} = \{\mathcal{X}(j)\}_{j=1,\ldots,t_{Ni},\ldots,t_{MC}}$$

with j the MCMC sampling steps, t_{Ni} the number of burn-in iterations of the sampler and t_{MC} the total number of MCMC iterations.

In order to infer the estimation for $\boldsymbol{\theta}$, the *maximum a posteriori* (MAP) estimator is adopted[4]. Considering the full posterior (10), one can obtain the marginal distribution $p(\boldsymbol{\theta}|\boldsymbol{v})$ by integrating out the hyperparameters $\boldsymbol{\pi}, \boldsymbol{\alpha}$ and α_0.

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{v}) &\propto \int p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{\pi}, \alpha_0, \boldsymbol{v}) d\boldsymbol{\alpha} d\boldsymbol{\pi} d\alpha_0 \\ &\propto \left(d + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2\right)^{-c-\frac{M}{2}} \prod_{i=1}^{N} \frac{\Gamma(a + \frac{\|\theta_i\|_0}{2}) B(\tilde{e}_i, \tilde{f}_i)}{\left(b + \frac{\theta_i^2}{2}\right)^{a+\frac{\|\theta_i\|_0}{2}}} \end{aligned} \tag{16}$$

where $\tilde{e}_i = e^{\langle q \rangle} + \|\theta_{\mathcal{J}_{i,k,\odot}}\|_0$ and $\tilde{f}_i = f^{\langle q \rangle} + |\mathcal{J}_{i,k,\odot}| - \|\theta_{\mathcal{J}_{i,k,\odot}}\|_0$.

Therefore, the MAP estimator of $\boldsymbol{\theta}$ can be computed by retaining the sample maximize the posterior distribution (16)

$$\hat{\boldsymbol{\theta}} \approx \arg\max_{\boldsymbol{\theta} \in \Theta} p(\boldsymbol{\theta}|\boldsymbol{v}) \tag{17}$$

where $\Theta = \{\boldsymbol{\theta}(j)\}_{j=t_{Ni},\ldots,t_{MC}}$.

Apparently, calculating the posteriors of each parameters, respectively, (12), (13), (14) and (15), only requires $O(N)$ multiplications. Therefore, the complete complexity for the Algorithm 1 can be easily calculated, which is $O(N^2)$.

[4] The minimum mean square estimator (MMSE) is not appropriate in this case, since averaging the simulated sparse samples may lead to non-sparse MMSE estimation.

4. Experiments

Conveniently, we denote the proposed algorithm CluSS, abbreviation of *Clustered Sparse Solver.* Then the following experiments are using the same settings, where the model hyperparameters for the priors in CluSS are set as follows: $a = b = c = d = 10^{-6}$, $(e^{\langle 0\rangle}, f^{\langle 0\rangle}) = (\frac{1}{M}, 1-\frac{1}{M}) \times |\mathcal{J}_{i,k,\odot}|$, $(e^{\langle 1\rangle}, f^{\langle 1\rangle}) = (\frac{1}{M}, \frac{1}{M}) \times |\mathcal{J}_{i,k,\odot}|$, $(e^{\langle 2\rangle}, f^{\langle 2\rangle}) = (1 - \frac{1}{M}, \frac{1}{M}) \times |\mathcal{J}_{i,k,\odot}|$, where $k = 1$, the initial conditions are set to $\alpha_i(0) = 1, \pi_i(0) = 0$, and $\alpha_0(0) = 1/\mathrm{var}(\boldsymbol{v}) \times 10^2$ for all $i \in \mathcal{V}_N$.

Then several experiments considered widely in CS literatures are implemented via CluSS, and comparisons are made to the state-of-the-art CS algorithms, respectively, Basis Pursuit (BP) [31], CoSaMP [32], Block-CoSaMP [10], (K, S)-sparse recovery algorithm via Dynamic Programming (Block-DP) [12] and Bayesian Compressive Sensing (BCS) [21]. Without special explanation, the sensing matrix $\boldsymbol{\Phi}$ is constructed randomly as in the seminal work [2], i.e., entries are drawn independently from Gaussian distribution $\mathcal{N}(0, 1/\sqrt{M})$.

4.1. General View

The objective of this subsection is to give an overall viewpoint for the proposed CluSS. Firstly, synthetic cluster structured sparse signals with length $N = 100$ are randomly generated. The sparsity S is set to 30 and the nonzero entries are set to ±1 uniformly distributed (or values drawn from a Gaussian distribution $\mathcal{N}(0, 1)$) and clustered into $K = 2$ blocks, see Fig. 3.

We first implement reconstruction via the aforementioned algorithms with noise free measurements on both ±1 spikes and Gaussian distributed spikes, where only $M = 50$ measurements are available. And the signal model parameters, such as sparsity and clusters, are optimally given: sparsity $\hat{S} = S = 30$, number of cluster $\hat{K} = K = 2$ and size of cluster $\hat{J} = 15$. Fig. 3 demonstrates the reconstruction results via BP (c), CoSaMP (d), Block-CoSaMP (e), BCS (f), CluSS (g) and Block-DP (h). The relative error of reconstruction is calculated by $e = \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 / \|\boldsymbol{\theta}\|_2$, where $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ are the estimated and the true coefficient vectors, respectively. It is shown that only CluSS can well recover the original cluster structured sparse signal and as an auxiliary, the evolution of the mixing weight $\boldsymbol{\pi}$ is given in Fig. 4.

4.2. Convergence diagnostic

When using the MCMC technique, the convergence diagnostic should be carried out to well determine the burn-in period. In this place, we use the Potential Scale Reduction Factor (PSRF) [33] (*Multivariate PSRF, MPSRF* [34] for multiple variables) to monitor the convergence of iterative simulations. The evolution of the PSRF can be shown in Fig. 5. Experimentally, the PSRF converges less than 1.5 when the sparse signal can be correctly reconstructed. Therefore, during the Gibbs sampling procedure of the following sections, the burn-in period is set to 250 iterations and then followed by a 50-sample-collection period.

4.3. Successful reconstruction rate versus sparsity

In this subsection, the objective is to compare the recovery abilities of the aforementioned algorithms for different oversampling rate (related to sparsity and measurements), denoting $\rho = \frac{S}{M}$. The size of CS problem is fixed with signal length $N = 100$ and measurement number $M = 50$, and the sensing matrix $\mathbf{\Phi}$ is generated as described at the beginning of this section. In the simulation, vary sparsity S from 1 to M with step 1, and then for each sparsity level, randomly generate 100 trials of cluster structured sparse signals with length N and blocks $K = 2$ (or $K = 4$). After that, the CS measurements are captured (noise free) through projecting the randomly generated sparse signal $\boldsymbol{\theta}$ on sensing matrix $\mathbf{\Phi}$. BP, Block-CoSaMP, CoSaMP, BC-S and CluSS are respectively exploited to carry out the CS reconstruction, where the required parameters for some of the algorithms are optimally set to $\hat{S} = S, \hat{K} = K$ and the size of clusters $\hat{J} = \lfloor S/K \rfloor$. The successful reconstruction is determined by the relative error between the true signal and its estimation, saying success if $e < 10^{-2}$ and fail for else. At last, the successful rate can be calculated through the ratio of total number of success events over the total number of trials, and the results are depicted in Fig. 7.

It is shown in Fig. 7 that CluSS has the highest rate when oversampling rate is approximately larger than 0.3. Moreover, CluSS has a satisfactory successful rate when $\rho \in [0.45, 0.5]$ while BP, Block-CoSaMP, CoSaMP and BCS has no hope to solve the CS problem successfully. On the other hand, compare the successful rate curve between $K = 4$ and $K = 2$, it is also shown that the fewer the clusters, the higher the successful rate for CluSS [5].

[5] It is worth noting that the reason of the unsatisfactory performance for Block-CoSaMP is that the blocks in the cluster structured sparse signals are not with identical size and fixed location. While the Block-DP is not compared here since it is very inefficient ($\mathcal{O}(N^3S^2K^2)$).

4.4. Robustness to noise

In this subsection, we will evaluate the robustness of the proposed algorithm. For a sparse signal with length $N = 100$, sparsity $S = 25$ and cluster $K = 2$, only $M = 50$ measurements are captured and then contaminated by Gaussian noises with variance σ_0 ranging from 0.01 to 0.1, namely, the SNR (signal to noise ratio) approximately ranging from 34dB to 12dB (see Fig. 6). The proposed algorithm CluSS and the other state-of-the-art algorithms can be used to recovery the original sparse signal. Run this experiment 100 times, then we can obtain the mean and the variance of the recovery SNR for each noise level, as shown in Fig. 6. The result shows that only CluSS can recover the sparse signal with acceptable error, which is in accordance with the result shown in Fig. 7.

4.5. Reconstruction with mismatch sparsity model

The state-of-the-art CS algorithms, such as Block-CoSaMP, Block-DP, etc. are designated to cope with the CS recovery problem with cluster prior. Nevertheless, they are only implementable for special cases, for instance, with fixed cluster locations or known number of clusters. In other words, a mismatch model (or unknown model parameters) will ruin the reconstruction performance for them. Oppositely, CluSS is nonparametric and hence more robust to mismatch sparsity model than the state-of-the-art CS algorithms specific for clustered sparse signals.

In this experiment, we generate a length $N = 100$ synthetic sparse signal with sparsity $S = 13$ and clusters $K = 2$, see Fig. 8(a). Only $M = 50$ measurements are obtained without noise. The comparisons are made to

CoSaMP, Block-CoSaMP and Block-DP, and the experiments are implemented with both the correct clustered sparsity model, where the model parameters are set to $\hat{S} = 13, \hat{K} = 2$ and the size of clusters $\hat{J} = 7$, and the incorrect clustered sparsity model, where $\hat{S} = 11, \hat{K} = 1$ and the size of clusters $\hat{J} = 11$. As shown in Fig. 8(c)(e)(g), with correct model parameters, all algorithms are capable to obtain the exact reconstruction, however, as shown in Fig. 8(d)(f)(h), with incorrect model parameters, only CluSS can reconstruct the true signal (see Fig. 8(b)).

4.6. Real Musical Signals

The above experiments are oriented to synthetic cluster structured sparse signals, where it is shown that CluSS can well preserve the cluster structures. In order to carry out the experiments in realistic applications, the object signals should exhibit as clustered blocks or possess the cluster structured sparse representations. In this place, we exploit soft musical signals which are not with complicated harmonics, and hence this kind of signals possess the cluster sparse representations in frequency domain [35, 27]. Fig. 9 gives a clip of this kind of musical signal (*Mozart*) played by the flute and its spectrum.

Setting frame size $N = 128$ and the number of measurements per frame $M = 60$ (or 80, 100), one can construct a sensing matrix $\mathbf{\Phi}$ following the method described at the beginning of this section. Then the measurements are captured by projecting each frame of the musical signal on the sensing matrix $\mathbf{\Phi}$. Then CluSS and the aforementioned algorithms are exploited to implement the reconstruction procedure [6]. The average means (Mean) of Relative Reconstruction Error (RRE) over all frames for each of algorithms are given in Tab. 1, as well as the corresponding standard variances (Std). The best recovery results are highlighted in bold type. It is shown that for different number of measurements M, the proposed CluSS is always with the best performance over the competitors.

On the other hand, in order to depict the property of CluSS in preserving cluster property, the zoom in of the spectrums of the reconstructions for each of algorithms are given in Fig. 11. In Fig. 11(a) and (b), only $M = 80$ (or $M = 100$ for (b)) measurements per frame are captured, and the spectrums of the reconstructions for each of algorithms apparently show the cluster preserving property for CluSS, while the competitors give worse results. Although Block-CoSaMP also has the property of preserving cluster structures, the optimal parameter for Block-CoSaMP cannot be well given, which results in unsatisfactory reconstructions for Block-CoSaMP.

4.7. An example of 2 dimensional cluster structured sparse signals

In this section, we extend CluSS to the 2 dimensional case, where only 1st-neighborhood is considered for the sparsity pattern, i.e., 4 neighbors for each pixel. The experiment is carried out with some patches of letters "M", "B","C" and "S" (16×16) with black background, thus it is with a lot of zero pixels in each patch and the nonzero pixels are clustered in blocks, as shown in Fig. 10. Then, only $M = 100$ noisy measurements (SNR$\approx$ 20dB) are obtained. The comparisons of performance are made to BP, CoSaMP

[6] The Fourier transform is employed to obtain the sparse representation.

and BCS. The results are shown in Fig. 10, where we can find that only CluSS gives cognizable reconstructions.

5. Conclusion

In this paper, we propose a new algorithm to recover the cluster structured sparse signals. Particularly, both *sparse prior* and *cluster prior* are taken into account via a hierarchical Bayesian model. MCMC sampling is proposed to implement the Bayesian inference. Unlike the existing recovery algorithms for *clustered sparsity model*, the proposed algorithm needs none of the parameters tuned manually, i.e. it is completely automatic. The experimental results show that the proposed algorithm is outstanding the state-of-the-art recovery algorithms for the recovery of cluster structured sparse signals.

References

[1] E. J. Candès and T. Tao. Decoding by Linear Programming. *IEEE T. Inform. Theory.*, 51(12):4203–4215, Dec. 2005.

[2] E. J. Candès and M. B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal. Proc. Mag.*, 25(2):21–30, March 2008.

[3] R. G. Baraniuk. Compressive Sensing [Lecture Notes]. *IEEE Signal. Proc. Mag.*, 24(4):118–121, July 2007.

[4] D. L. Donoho. Compressed Sensing. *IEEE T. Inform. Theory.*, 52(4):1289–1306, April 2006.

[5] E. J. Candès and T. Tao. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *IEEE T. Inform. Theory.*, 52(12):5406–5425, Dec. 2006.

[6] T. Blumensath, M. Yaghoobi, and M. E. Davies. Iterative Hard Thresholding and ℓ_0 Regularisation. In *ICASSP*, 2007.

[7] R. Chartrand and Wotao Yin. Iteratively Reweighted Algorithms for Compressive Sensing. In *ICASSP*, 2008.

[8] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[9] T. Blumensath and M. E. Davies. Sampling Theorems for Signals From the Union of Finite-Dimensional Linear Subspaces. *IEEE T. Inform. Theory.*, 55(4):1872–1882, 2009.

[10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-Based Compressive Sensing. *IEEE T. Inform. Theory.*, 56(4):1982–2001, 2010.

[11] Volkan Cevher, Chinmay Hegde, Marco F. Duarte, and Richard G. Baraniuk. Sparse Signal Recovery Using Markov Random Fields. In *NIPS*, 2008.

[12] Volkan Cevher, Piotr Indyk, Chinmay Hegde, and Richard G. Baraniuk. Recovery of Clustered Sparse Signals from Compressive Measurements. In *Int. Conf. on Sampling Theory and Applications (SAMPTA)*, 2009.

[13] Y. C. Eldar, P. Kuppinger, and H. Bolcskei. Block-Sparse Signals: Uncertainty Relations and Efficient Recovery. *IEEE T. Signal. Proces.*, 58(6):3042–3054, 2010.

[14] L. He and L. Carin. Exploiting Structure in Wavelet-Based Bayesian Compressive Sensing. *IEEE T. Signal. Proces.*, 57(9):3488–3497, 2009.

[15] J. A. Tropp and A. C. Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE T. Inform. Theory.*, 53(12):4655–4666, Dec. 2007.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[17] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[18] V. Cevher, P. Indyk, L. Carin, and R. G. Baraniuk. Sparse Signal Recovery and Acquisition with Graphical Models. *IEEE Signal. Proc. Mag.*, 27(6):92–103, 2010.

[19] Michael E. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[20] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Bayesian Compressive Sensing Using Laplace Priors. *IEEE T. Image. Process.*, 19(1):53–63, 2010.

[21] S. Ji, X. Ya, and L. Carin. Bayesian Compressive Sensing. *IEEE T. Signal. Proces.*, 56(6):2346–2356, June 2008.

[22] S. Ji, D. Dunson, and L. Carin. Multitask Compressive Sensing. *IEEE T. Signal. Proces.*, 57(1):92–106, 2009.

[23] J. Kormylo and J. Mendel. Maximum Likelihood Detection and Estimation of Bernoulli-Gaussian Processes. *IEEE T. Inform. Theory.*, 28(3):482–488, 1982.

[24] J. Idier and Y. Goussard. Stack Algorithm for Recursive Deconvolution of Bernoulli-Gaussian Processes. *IEEE T. Geosci. Remote.*, 28(5):975–978, 1990.

[25] A. Doucet and P. Duvaut. Bayesian Estimation of State-space Models Applied to Deconvolution of Bernoulli-Gaussian Processes. *Signal Processing*, 57(2):147–161, 1997.

[26] C. Févotte and S.J. Godsill. A Bayesian Approach for Blind Separation of Sparse Sources. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2174–2188, 2006.

[27] C. Févotte, B. Torrésani, L. Daudet, and S.J. Godsill. Sparse Linear Regression with Structured Priors and Application to Denoising of Musical Audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):174–185, 2008.

[28] N. Dobigeon and J.Y. Tourneret. Bayesian Orthogonal Component Analysis for Sparse Representation. *IEEE T. Signal. Proces.*, 58(5):2675–2685, 2010.

[29] L. Chaâri, J.C. Pesquet, J.Y. Tourneret, P. Ciuciu, and A. Benazza-Benyahia. A Hierarchical Bayesian Model for Frame Representation. *IEEE T. Signal. Proces.*, 58(11):5560–5571, 2010.

[30] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, 2004.

[31] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1999.

[32] D. Needell and J.A. Tropp. CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples. *Applied and Computational Harmonic Analysis*, 26(3):301 – 321, 2009.

[33] A. Gelman and D.B. Rubin. Inference from Iterative Simulation using Multiple Sequences. *Statistical science*, 7(4):457–472, 1992.

[34] S.P. Brooks and A. Gelman. General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.

[35] G. Yu, S. Mallat, and E. Bacry. Audio Denoising by Time-Frequency Block Thresholding. *IEEE T. Signal. Proces.*, 56(5):1830–1839, 2008.

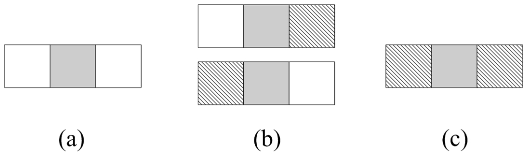


Figure 1: Three different cluster pattern for 1D signals.

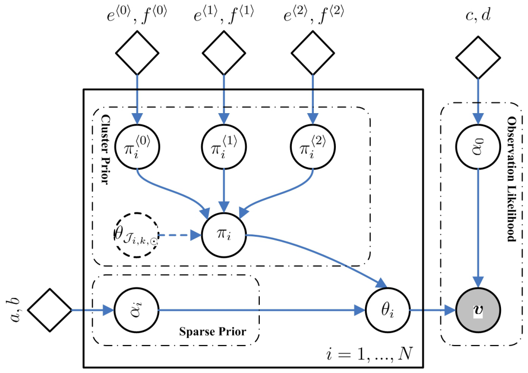
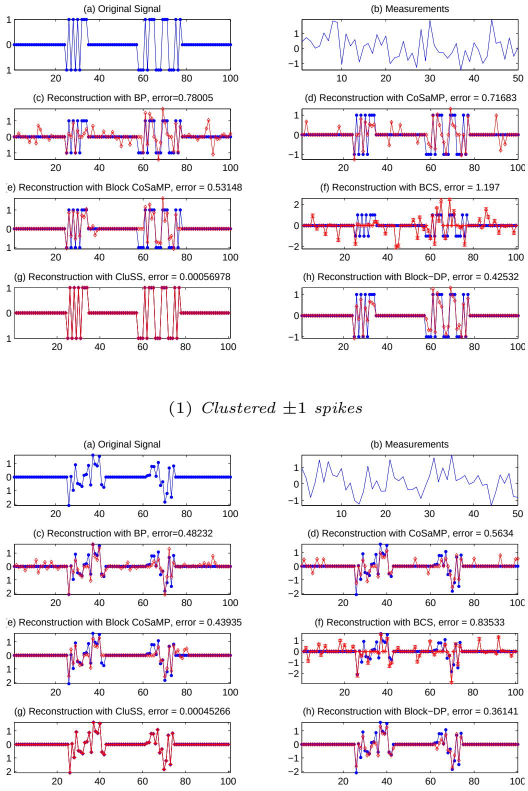

$e^{\langle 0 \rangle}, f^{\langle 0 \rangle}$
$e^{\langle 1 \rangle}, f^{\langle 1 \rangle}$
$e^{\langle 2 \rangle}, f^{\langle 2 \rangle}$
c, d
Cluster Prior
$\pi_i^{\langle 0 \rangle}$
$\pi_i^{\langle 1 \rangle}$
$\pi_i^{\langle 2 \rangle}$
α_0
Observation Likelihood
$\theta_{\mathcal{J}_{i,k,\odot}}$
π_i
a, b
α_i
θ_i
v
Sparse Prior
$i = 1, ..., N$

Figure 2: Hierarchical Bayesian Generative Model for Sparse Signal.

Table 1: Relative Reconstruction Error (RRE) of Musical Signals

Meas.	RRE	BP	CoSaMP	Block-CoSaMP	BCS	CluSS
$M = 100$	Mean	0.0076	0.0330	0.0564	0.0068	**0.0022**
	Std	0.0073	0.0237	0.0337	0.0195	**0.0021**
$M = 80$	Mean	0.0523	0.0728	0.0993	0.0552	**0.0195**
	Std	0.0397	0.0685	0.0623	0.0527	**0.0191**
$M = 60$	Mean	0.1690	0.2548	0.2489	0.2126	**0.1050**
	Std	0.1051	0.1384	0.1702	0.1929	**0.0904**



(1) *Clustered* ±1 *spikes*

(2) *Clustered Gaussian spikes*

Figure 3: Reconstruction of (1) *clustered* ±1 *spikes* and (2) *clustered Gaussian spikes* for $N = 100, M = 50, S = 30, K = 2$ with noise free measurements.

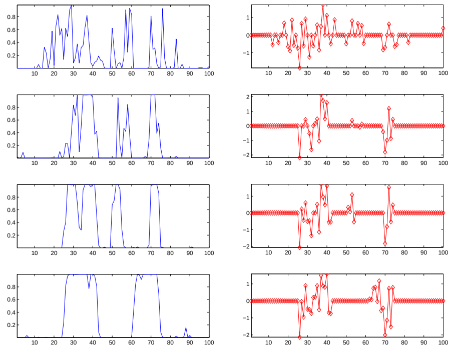


Figure 4: Evolution of the mixing weight $\boldsymbol{\pi}$ (left row) and sparse signal $\boldsymbol{\theta}$ (right row), respectively at 1, 10, 40 and 100 iteration(s).

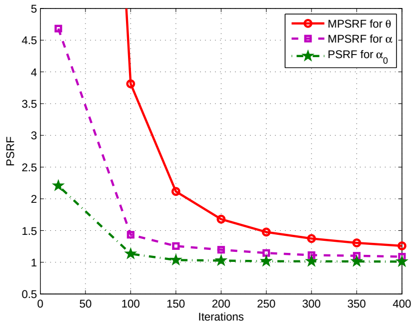


Figure 5: Evolution of MPSRF for sparse signal $\boldsymbol{\theta}$ and inverse variance $\boldsymbol{\alpha}$ of sparsity model, and PSRF for inverse noise variance α_0;

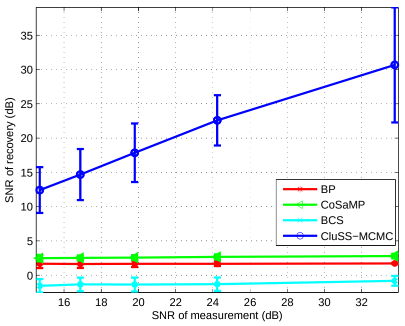


Figure 6: Robustness to different level of noise.

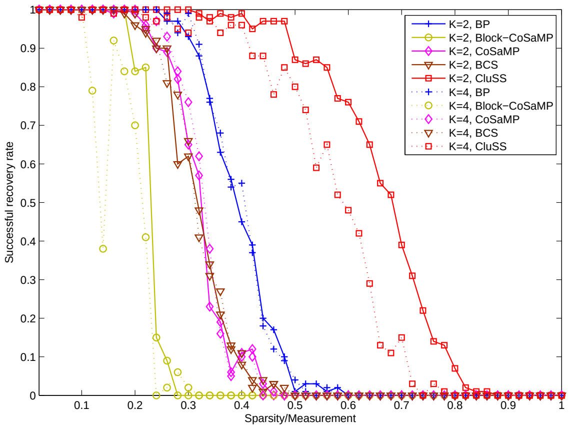
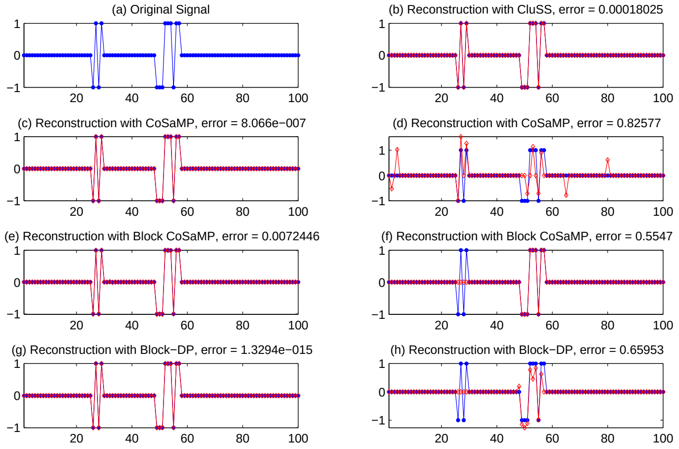


Figure 7: The successful recovery rate for different sparsity over measurements.


(a) Original Signal

(b) Reconstruction with CluSS, error = 0.00018025

(c) Reconstruction with CoSaMP, error = 8.066e−007

(d) Reconstruction with CoSaMP, error = 0.82577

(e) Reconstruction with Block CoSaMP, error = 0.0072446

(f) Reconstruction with Block CoSaMP, error = 0.5547

(g) Reconstruction with Block−DP, error = 1.3294e−015

(h) Reconstruction with Block−DP, error = 0.65953

Figure 8: The influence of sparse model parameters on the reconstruction of *clustered spikes* ±1, where $N = 100, M = 50, S = 13, K = 2$.

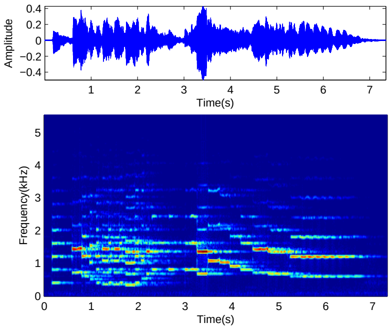


Figure 9: The original musical signal and its spectrum.

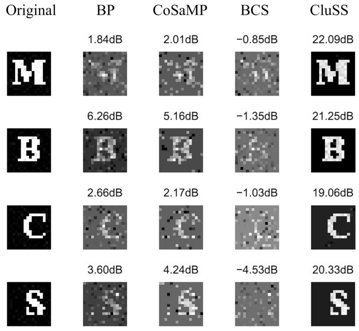
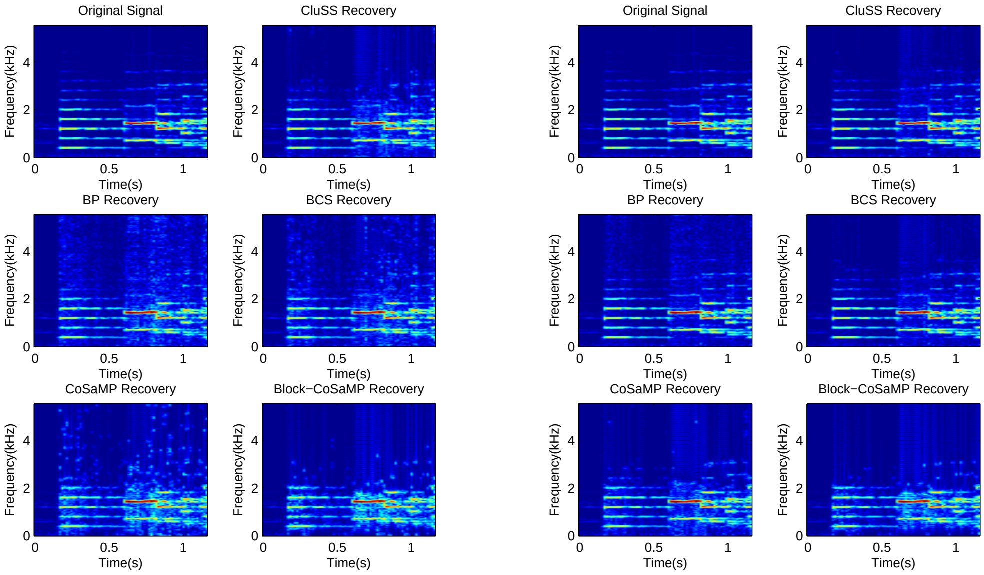


Figure 10: An example of 2 dimensional cluster structured sparse signals.



(a) 80 measurements

(b) 100 measurements

Figure 11: The spectrum of reconstructions via different algorithms.