A Framework for Clustered and Skewed Sparse Signal Recovery

Sheng Wang, *Student Member, IEEE*, and Nazanin Rahnavard, *Member, IEEE*

***Abstract*—A novel framework, clustered-skew normal mixture-belief propagation, is developed to solve the reconstruction of undersampled clustered signals, where the magnitudes of signal coefficients in each cluster are distributed asymmetrically w.r.t the cluster mean. To address the skewness feature, a finite skew-normal density mixture is utilized to model the prior distribution, where the marginal posterior of the signal is inferred by an efficient approximate message-passing-based algorithm. An expectation-maximization-based algorithm is developed to estimate the mixture density. The clustered property is then modeled by the Potts model, and a loopy belief propagation algorithm is designed to promote the spatial feature. Experimental results show that our technique is highly effective and efficient in exploiting both the clustered feature and asymmetrical feature of the signals and outperforms many sophisticated techniques.**

***Index Terms*—Compressed sensing, asymmetrical signal, approximate message passing, expectation-maximization algorithms.**

I. Introduction

COMPRESSIVE sampling is a paradigm to solve for the correct target signal $\underline{x} \in \mathbb{R}^{N\times 1}$, of the under-determined linear system,

$$\underline{y} = \mathbf{A}\underline{x} + \underline{e}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a known random sampling matrix with $M \ll N$, $\underline{y} \in \mathbb{R}^{M\times 1}$ is the random measurement, and $\underline{e} \in \mathbb{R}^{M\times 1} \sim \mathcal{N}(\underline{0}, \sigma_e^2 \mathrm{I}_{M\times M})$ is the measurement white Gaussian noise.

Without any additional knowledge of the signal, solving for the correct $\underline{x}$ is ill-posed, and there can be infinitely many solutions satisfying (1). What compressive sensing theory [1]–[4] states is that, reliable reconstruction of the signal from the under-determined system (1) is possible, provided that $\underline{x}$ is adequately sparse, and the sampling matrix $\mathbf{A}$ satisfies the so called *Restricted Isometry Property* [1]. Here by sparse, it is intended that the energy of the signal is primarily carried by $Q \ll N$ coefficients of $\underline{x}$, which is referred to as significant coefficients, whereas the energy of the rest coefficients, i.e., insignificant coefficients, are inconsequential.

The advantage of compressive sampling in solving an under-determined system makes it attractive in fields where increasing sampling rate is costly, and a great number of applications have been inspired. For instance, Compressive Sampling has found great applications in remote sensing [5], medical imaging [6], wireless communication system [7], wireless sensor networks [8], [9], multimedia processing [10]–[12], and anomaly detection [13], [14].

Reconstruction of clustered sparse signal is an attractive topic of compressive sensing community. For example, in multimedia processing [11], it is found that significant pixels of video difference frames tend to form clusters, due to the temporal redundancy of consecutive video frames. Another promising application can be found in sensor networks to detect abnormal environment events [15], where in the presence of abnormality, sensors close to the event give significant and correlated outputs, while those outside the scope of the event return outputs resembling the no-event average.

Many sophisticated strategies have been proposed to exploit the clustered property in compressive sensing tasks. In [16], a pruning stage is designed to encourage clustered property based on Orthogonal Matching Pursuit, and the developed method is referred to as Structural Orthogonal Matching Pursuit (*SOMP*). In [17], a Markov Chain Monte Carlo strategy is employed to solve the compressive sensing of clustered structured sparse signals (*CluSS*), and the developed method turns out to realize faithful reconstruction in dealing with block clustered sparse signals. In [11], a Structural Re-weighted ℓ_1 norm minimization technique (*SRL1*) is developed, where signal coefficients are allocated with weights determined by the magnitudes of their corresponding neighbors. In [18], a Lattice Matching Pursuit (*LaMP*) is developed, where the clustered sparsity of the signal is modelled by the Ising model, with which the signal support is estimated and the reconstruction is directed. A Pattern-Coupled Sparse Bayesian Learning algorithm (*PCSBL*) is developed in [48], and the clustered feature is exploited by a pattern-coupled hierarchical Gaussian prior model and generalized approximate message passing.

Compressive sensing of asymmetrical signals is another line of research, and signals of this type can be found in Multi-Input Multi-Output (MIMO) wireless communication systems [22], and weather sensor networks [7], [15], [23]. In [24],

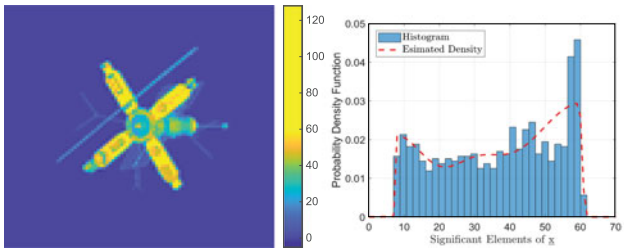
Manuscript received October 5, 2017; revised April 1, 2018; accepted May 1, 2018. Date of publication May 22, 2018; date of current version June 22, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xin Wang. This work was supported by the National Science Foundation under Grant CCF-1718195. *(Corresponding author: Sheng Wang.)*

S. Wang is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: sheng.wang@okstate.edu).

N. Rahnavard is with the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: nazanin@eecs.ucf.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2839622



(a) (b)

Fig. 1. Satellite Image: clustered property and asymmetrical features (a) Satellite image (b) Histogram and estimated density of significant coefficients using skew-normal distribution mixture.

A *Bernoulli* non-negative *Gaussian* mixture is employed to model the distribution of sparse signals with non-negative coefficients, and an efficient approximate message passing based algorithm is proposed. An effective framework is proposed in [7] to deal with sparse signals with skewness feature, where a two-state normal and skew normal mixture density is utilized to model the prior distribution of the signals. The asymmetrical feature is captured by the skew normal density component, and the signal is estimated by an approximate message passing based algorithm.

In this work, we move one step further by approaching the compressive sensing of clustered sparse signals, where the magnitudes of each cluster are distributed asymmetrically about the corresponding cluster mean. One motivating example can be found in the *Satellite* image [48] shown in Fig. 1(a), where the histogram of the significant coefficients of the image and estimated distribution[1] are shown in Fig. 1(b). Specifically, the clustered feature lies in the fact that structurally, Fig. 1(a) contains 4 of yellow-to-orange *rectangles* which correspond to the *legs* of *Satellite*, a green-to-yellow *trapezoid* in the center which corresponds to the *body*, a number of light-blue *poles*, and the dark-blue background. Besides, the asymmetrical feature can be found in the histogram Fig. 1(b), where the pixel intensities of *poles* are distributed *right-skewed* about 10, whereas the intensities of *legs* are distributed *left-skewed* about 60.

To get a faithful reconstruction of the signals, we adopt a *divide-and-conquer* methodology, and decompose the task into three modules.

First of all, to address the skewness feature, a finite skew-normal distribution mixture is utilized to model the prior distribution of the signal. Skew normal distribution [25] generalizes normal distribution, and is more flexible in dealing with asymmetric features. An efficient approximate message passing algorithm, which takes the mixture distribution and the hidden states of signal coefficients as inputs, is designed to iteratively derive the marginal posterior and the *Minimal-Mean-Squared-Error* (MMSE) estimate of the signal by propagating local beliefs between $\underline{x}$ and $\underline{y}$.

Subsequently, following the approximate message passing module, an Expectation-Maximization-based algorithm is developed to estimate the mixture density from the MMSE estimate of the signal. The number of mixture components is estimated in an efficient and non-parametric way.

[1]Estimated using our proposed skew-normal density mixture model.

Moreover, given the MMSE and the mixture density estimates from previous modules, a loopy message passing based algorithm is designed, where the compatibility of neighboring coefficients is regularized by the *Potts* model, after which the hidden states of signal coefficients can be estimated, and the clustered property can be promoted.

Overall, our proposed technique, referring to as *CL-SNM-BP*, alternates among exploiting the measurement, drawing inference of the finite mixture model, and taking advantage of the clustered property. These three modules work sequentially and iteratively, after which a refined reconstruction of the signal can be obtained.

To the best of our knowledge, our method is among the first few works taking both asymmetry and clustered sparsity into account in compressive sensing tasks. Compared to [7] which has analyzed general asymmetrical sparse signals, our developed technique is designed to exploit the clustered features on top of asymmetry. Moreover, compared to the two-states mixture model [7] with fixed location parameters, our technique utilizes a finite mixture model, which allows for multiple skew normal distribution components with arbitrary location parameters, and can therefore accommodate more general signals.

Existing studies [18], [26] utilize Markov random field and *Ising* model [27] to exploit the clustered property. While being highly effective in recovering the support sets of signals, they are incapable of discriminating diverse hidden states of significant coefficients. Taking advantage of the *Potts* model, our developed method not only promotes clustered property, but is also adequately responsive to different hidden states of signal coefficients. Therefore, compared to existing methods, clustered property is exploited in a more informative way.

The remainder of this paper is organized as follows. The signal model and the framework of our proposed technique are introduced in Section II. Approximate message passing employing the skew normal mixture prior is detailed in Section III. In Section IV, an Expectation-Maximization based algorithm is put forward to infer the finite skew normal density mixture. The hidden states estimate using loopy message passing and *Potts* model is derived in Section V. Simulation results are summarized in Section VI, and Section VII concludes the work.

The following notations are used throughout the paper. Bold symbols denote matrices exclusively, and symbols with underline denote vectors. $\phi(\cdot)$ and $\Phi(\cdot)$ are reserved for standard normal probability density function (*pdf*) and cumulative distribution function (*cdf*). $\mathcal{SN}(\cdot)$ is reserved for the *pdf* of skew normal density.

II. Signal Model and Problem Definition

A. Signal Model

1) Signal Representations: Denote the two dimensional signal $\mathbf{x} = (x_{ij}) \in \mathbb{R}^{d\times d}$ as the outcome of random variable $\mathbf{X} = (X_{ij}) \in \mathbb{R}^{d\times d}$, where $1 \leq i, j \leq d$, and $d^2 = N$. For ease of notation, in this work, the two dimensional signal $\mathbf{x}$ is also represented as a one dimensional column vector, $\underline{x} = [x_1, \ldots, x_n, \ldots, x_N]^\intercal$, where x_{ij} is mapped to x_n in one dimensional form with $n = (i-1)\times d + j$, and $1 \leq n \leq N$. Similarly,

$\underline{X} = [X_1, \ldots, X_n, \ldots, X_N]^\intercal$ is the one dimensional representation of $\mathbf{X}$.

It is also convenient to represent the signal as a concatenation of clusters. Specifically, let G be the total number of clusters, out of which, $0 \leq G_s < G$ clusters are significant, with the remaining being insignificant. Therefore, the signal can be written as, $\underline{x} = [\underline{x}_1^\intercal, \ldots, \underline{x}_g^\intercal, \ldots, \underline{x}_G^\intercal]^\intercal$, with $\underline{x}_g = [x_{g(1)}, \ldots, x_{g(d_g)}]^\intercal$ denoting the g-th cluster, where $1 \leq g \leq G$. Besides, d_g denotes the cardinality of cluster g, and $\sum_{g=1}^{G} d_g = N$.

In this work, it is assumed that signals are drawn from a probabilistic density ensemble of K density components. Let $S_n \in \{1, \ldots, K\}$ be a random variable indicating the corresponding state of signal coefficient X_n, and denote $\underline{S} = [S_1, \ldots, S_N]^\intercal \in \mathbb{R}^{N\times 1}$ as the state random vector, with the corresponding realization $\underline{s} = [s_1, \ldots, s_N]^\intercal$ being the state vector.

Without any constraint, the state vector $\underline{s}$ lies in the $\{1, \ldots, K\}^N$ subspace of $\mathbb{R}^N$. To realize clustered property, we restrict the states within a cluster to be homogenous, i.e., $s(i) = s(j)$ for any $x_i, x_j \in \underline{x}_g$.

Additionally, let $\mathbf{V} = (V_{nk}) \in \mathbb{R}^{N\times K}$ be the state probability matrix, where V_{nk} denotes the probability of X_n taking state k, with the non-negative probability constraint $0 \leq V_{nk} \leq 1$, and unitary row sum constraint $\sum_{k=1}^{K} V_{nk} = 1$.

2) Skew Normal Density: Skew normal density is a continuous probabilistic distribution generalized from the normal distribution [25],

$$\mathcal{SN}(X = x|\xi, \omega, \alpha) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right), \quad (2)$$

where ξ, ω and α denote location, scale and shape parameters, respectively. As can be seen in (2), compared to the normal distribution, the shape parameter α allows for a non-zero skewness,[2] thus enables to capture the skewness feature of signals.

3) Mixture Density Model: In this work, it is assumed that for any cluster g, the states of its coefficients are homogeneous, and the coefficients are jointly independent, conditioned on the states, i.e.,

$$p\left(\underline{X}_g = \underline{x}_g|S(\underline{x}_g) = k\right) = \prod_{x\in\underline{x}_g} \mathcal{SN}(x|\xi_k, \omega_k, \alpha_k), \quad (3)$$

where skew normal density (2) is employed to represent the density of state $k \in 1, \ldots, K$.

Besides, the states of clusters are assumed to follow a K-state categorical distribution, e.g.,

$$p\left(S(\underline{x}_g) = k\right)\big) = \lambda_k, \quad (4)$$

where $\sum_{k=1}^{K} \lambda_k = 1$, and $\lambda_k \geq 0$.

Therefore, the *pdf* of the signal can be written as the following mixture,

$$f(\underline{x}; \mathbf{\Theta}) \sim \sum_{k=1}^{K} \lambda_k p(\underline{x}; \underline{\theta}_k), \quad (5)$$

where $\mathbf{\Theta} = [\underline{\theta}_1, \ldots, \underline{\theta}_K]^\intercal$ represents the parameter matrix, with $\underline{\theta}_k = [\xi_k, \omega_k, \alpha_k]$ denoting the parameter vector specifying the k-th skew-normal density component, and $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]^\intercal$ denoting the non-negative mixing weight vector.

[2]Skew normal distribution (2) becomes normal with $\alpha = 0$, and is left-skewed, and right-skewed when $\alpha < 0$, and $\alpha > 0$, respectively.

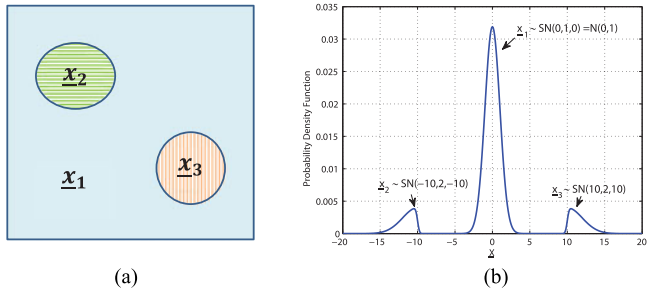


Fig. 2. Clustered sparse signal and skew normal mixture density (a) Signal with $G = 3$ clusters, where $G_s = 2$ clusters are significant. (b) Mixture density of $K = 3$ Skew Normal density components.

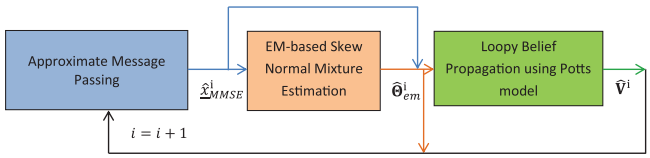


Fig. 3. Diagram of CL-SNM-BP.

Fig. 2(a) is a toy example of a clustered sparse signal, generated from the corresponding skew normal mixture density shown in Fig. 2(b). Following previous notations, the signal in Fig. 2(a) can be written as a concatenation of $G = 3$ clusters, i.e., $\underline{x} = [\underline{x}_1^\intercal, \underline{x}_2^\intercal, \underline{x}_3^\intercal]^\intercal$, where $\underline{x}_1^\intercal$ is an insignificant cluster (a cluster with insignificant data values), $\underline{x}_2^\intercal$ and $\underline{x}_3^\intercal$ are significant clusters. Besides, $\underline{x}_1$ is drawn from $p(\underline{x}_1; \underline{\theta}_1)$, with $\underline{\theta}_1 = [\xi_1 = 0, \omega_1 = 1, \alpha_1 = 0]^\intercal$, $\underline{x}_2$ is drawn from $p(\underline{x}_2; \underline{\theta}_2)$, with $\underline{\theta}_2 = [\xi_2 = -10, \omega_2 = 2, \alpha_2 = -10]^\intercal$, and $\underline{x}_3$ is drawn from $p(\underline{x}_3; \underline{\theta}_3)$, with $\underline{\theta}_3 = [\xi_3 = 10, \omega_1 = 2, \alpha_1 = 10]^\intercal$. The mixing weight in Fig. 2(b) is set to $\underline{\lambda} = [\lambda_1 = 0.8, \lambda_2 = 0.2, \lambda_3 = 0.2]$.

B. Problem Definition and System Architecture

We adopt a Bayesian perspective in the reconstruction phase of the compressive sensing task, with the goal being set to derive a faithful estimate of signal by maximizing the posterior distribution $p(\underline{x}|\underline{y}, \mathbf{V}, \mathbf{\Theta})$. As neither mixture parameters $\mathbf{\Theta}$ nor the state probability $\mathbf{V}$ is known, an effective algorithm is developed to seek a reliable reconstruction of the signal by iteratively applying the sub-modules shown in Fig. 3.

As can be seen in Fig. 3, at iteration i, CL-SNM-BP starts with an approximate message passing module, where a MMSE estimate of the signal is obtained by calculating the conditional expectation of the posterior, i.e., $\hat{\underline{x}}^i_{\text{MMSE}} = E[\underline{X}|\underline{Y} = \underline{y}, \mathbf{V}^{i-1}, \mathbf{\Theta}^{i-1}]$.

Subsequently, $\hat{\underline{x}}^i_{\text{MMSE}}$ is fed to the second module to get an estimate of the mixture density parameters $\underline{\Theta}$. In our technique, this is realized by seeking a maximum likelihood estimate (MLE) solution, $\widehat{\mathbf{\Theta}}^i_{\text{EM}} = \arg\max p\left(\underline{X} = \hat{\underline{x}}^i_{\text{MMSE}}|\mathbf{\Theta}\right)$, using an Expectation-Maximization-based method.

The last module involves estimating the probability state $\mathbf{V}$. Specifically, taking mixture density estimate $\widehat{\mathbf{\Theta}}^i_{\text{EM}}$, and the reconstruction of signal $\hat{\underline{x}}^i_{\text{MMSE}}$ as inputs, a loopy belief propagation based technique is set forth to infer the probability state, while promoting the clustered property.

The above completes the work flow of our technique. The proposed method, CL-SNM-BP, alternates between these modules and works in an iterative fashion, where at the end of iteration i, the state probability matrix $\widehat{\mathbf{V}}^i$ and the parameters of the skew normal mixture $\widehat{\mathbf{\Theta}}^i_{\text{EM}}$, are fed back to the approximate message passing module, and iteration $i+1$ starts.

III. Approximate Message Passing Employing Skew Normal Mixture Prior

In this section, to capture the skewness feature, we employ a finite skew normal density mixture (5) as the prior distribution of the signals. Given $\widehat{\mathbf{V}}^{i-1}$ and $\widehat{\mathbf{\Theta}}^{i-1}_{\text{EM}}$, an efficient approximate message passing algorithm is proposed to make inference of the signal by exchanging beliefs between variable nodes $\underline{x}$ and check nodes $\underline{y}$.

It is worthy noticing that, a similar technique can be found in [7], where a two-state normal and skew normal mixture is employed to model signals whose significant coefficients are skewed about the origin $x=0$. Our work here considers a multi-state skew normal mixture with arbitrary location parameters, and is capable of accommodating varying number of mixture components. Therefore, [7] can be viewed as a special case of our work.

A. Bayesian Inference by Approximate Message Passing

Approximate message passing [28], [29] is a powerful method enabling efficient and reliable Bayesian inference of the posteriors. In approximate message passing, the sampling process (1) is viewed as a bipartite factor graph, where $\underline{x}=[x_1,\ldots,x_n,\ldots,x_N]^{\mathsf{T}}$ is referred to as variable nodes, and $\underline{y}=[y_1,\ldots,y_m,\ldots,y_M]^{\mathsf{T}}$ is known as check nodes, with the entry A_{mn} being the edge connecting x_n and y_m.

The marginal posteriors are then estimated by iteratively exchanging local beliefs between variable nodes $\underline{x}$ and check nodes $\underline{y}$. Specifically, at iteration i, let $\nu^{(i)}_{x_n\to y_m}(x_n)$ denote the message from the variable node x_n to the check node y_m, and $\nu^{(i)}_{y_m\to x_n}(x_n)$ represent the message from the check node y_m to the variable node x_n, where

$$\nu^{(i)}_{x_n\to y_m}(x_n)=\mathcal{N}(x_n;\mu^{(i)}_{x_{nm}},\sigma^{2\,(i)}_{x_{nm}}), \tag{6}$$

$$\nu^{(i)}_{y_m\to x_n}(x_n)=\mathcal{N}(x_n;\mu^{(i)}_{y_{mn}},\sigma^{2\,(i)}_{y_{mn}}), \tag{7}$$

with the mean and variance being evaluated as,

$$\mu^{(i)}_{x_{nm}}=\int_{-\infty}^{\infty}x_n\nu^{(i)}_{x_n\to y_m}(x_n)\,\mathrm{d}x_n, \tag{8}$$

$$\sigma^{2\,(i)}_{x_{nm}}=\int_{-\infty}^{\infty}(x_n-\mu^{(i)}_{x_{nm}})^2\nu^{(i)}_{x_n\to y_m}(x_n)\,\mathrm{d}x_n, \tag{9}$$

$$\mu^{(i)}_{y_{mn}}=(y_m-\sum_{t\in[1,\ldots,N]\backslash\{n\}}A_{mt}\mu^{(i)}_{x_{tm}})/A_{mn}, \tag{10}$$

$$\sigma^{2\,(i)}_{y_{mn}}=(\sigma_e^2+\sum_{t\in[1,\ldots,N]\backslash\{n\}}A^2_{mt}\sigma^2_{x_{tm}})/A^2_{mn}. \tag{11}$$

Combining the skew normal mixture density prior (2) and (5), the message from x_n to y_m is updated in $(i+1)$-th iteration as,

$$\nu^{(i+1)}_{x_n\to y_m}(x_n)\cong\mathcal{N}(x_n;a^{(i)}_{nm},b^{2\,(i)}_n)\sum_{k=1}^{K}\lambda_k\mathcal{SN}(x_n;\xi_k,\omega_k,\alpha_k). \tag{12}$$

where

$$a^{(i)}_{nm}\triangleq\sum_{u\in[1,\cdots,M]\backslash\{m\}}A_{un}\mu^{(i)}_{y_{un}}, \tag{13}$$

$$b^{2\,(i)}_n\triangleq\frac{1}{M}\sum_{u\in[1,\cdots,M]}A^2_{un}\sigma^{2\,(i)}_{y_{un}}\quad. \tag{14}$$

It is noteworthy that (12) involves the product of normal density function and skew normal density function, i.e., $\mathcal{N}(x;a^{(i)}_{nm},b^{2\,(i)}_n)\mathcal{SN}(x;\xi_q,\omega_q,\alpha_q)$. A special case of this problem, where the location parameter is fixed to $\xi=0$, was studied in [7] for signals that are asymmetrical about the origin $x=0$. For arbitrary value of ξ, we come up with the following *Lemma 1* and *Lemma 2* to evaluate the corresponding statistics.

Lemma 1: Denote $\mathcal{SN}(x;\xi,\omega,\alpha)$ as the skew normal density with parameters being (ξ,ω,α), and let $\mathcal{N}(x;a,b^2)$ be the normal density function with mean value a and variance b^2, then the product $Z(a,b,\xi,\omega,\alpha)\times\mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2)$ is a probability density function, i.e., $Z(a,b,\xi,\omega,\alpha)\int_{-\infty}^{\infty}\mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2)\mathrm{d}x=1$, with

$$Z(a,b,\xi,\omega,\alpha)=\frac{\varsigma}{2\phi\left(\frac{a-\xi}{\varsigma}\right)\Phi(\eta)} \tag{15}$$

where $\varsigma=\sqrt{b^2+\omega^2}$, $\eta=\frac{\kappa+h\mu}{\sqrt{1+h^2\sigma^2}}$, $h=\frac{\alpha}{\omega}$, $\kappa=-h\xi$, $\mu=\frac{a\omega^2+\xi b^2}{\varsigma^2}$, and $\sigma^2=\frac{b^2\omega^2}{\varsigma^2}$.

Proof:

$$\mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2) \tag{16}$$

$$=\frac{2}{\omega b}\phi\left(\frac{x-a}{b}\right)\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) \tag{17}$$

$$=\frac{1}{\pi\omega\sigma}\exp\left(\frac{1}{2\sigma^2}\left(\mu^2-\frac{b^2\xi^2+\omega^2a^2}{\varsigma^2}-(x-\mu)^2\right)\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) \tag{18}$$

It is noticed that (18) involves $\Phi(\alpha\frac{x-\xi}{\omega})$, therefore, applying *Lemma 1* of [7], the above *Lemma 1* holds. ■

As a direct extension of *Lemma 3* in [7], the following *Lemma 2* is derived.

Lemma 2: Let a random variable X follows the distribution $X\sim Z(a,b,\xi,\omega,\alpha)\times\mathcal{N}(X;a,b^2)\mathcal{SN}(X;\xi,\omega,\alpha)$, then the mean $\mathrm{E}(X)$ is given by

$$\mathrm{E}(X)=\mu+\zeta\frac{\phi(\eta)}{\Phi(\eta)}, \tag{19}$$

and the variance is

$$\mathrm{Var}(X)=\mu^2+\sigma^2+\rho\zeta\frac{\phi(\eta)}{\Phi(\eta)}-\mathrm{E}^2(X), \tag{20}$$

where $\zeta=\frac{h\sigma^2}{\sqrt{1+h^2\sigma^2}}$, and $\rho=\frac{2\mu+\mu h^2\sigma^2-\kappa h\sigma^2}{1+h^2\sigma^2}$.

As a result of *Lemma 2*, and omitting the iteration superscript, (12) can be approximated by the normal density as,

$$\nu_{x_n \to y_m}(x_n) \cong \mathcal{N}(\mu_{x_{nm}}, \sigma^2_{x_{nm}}), \tag{21}$$

in which

$$\mu_{x_{nm}} = \mathbb{F}(a_{nm}, b_n^2, \mathbf{\Theta}, \mathbf{V}) = C_n \sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}} E_{nk}, \tag{22}$$

$$\begin{aligned}\sigma^2_{x_{nm}} &= \mathbb{G}(a_{nm}, b_n^2, \mathbf{\Theta}, \mathbf{V}) \\ &= \sum_{k=1}^{K} p_{nk}(E_{nk}^2 + Var_{nk}) - \left(\sum_{k=1}^{K} p_{nk} E_{nk}\right)^2,\end{aligned} \tag{23}$$

where E_{nk} and Var_{nk}^2 can be calculated as (19) and (20) with corresponding parameters $\kappa_{nm}^{(i)}, \varsigma_n^{(i)}, \xi_k$, ω_k and α_k of (12). It should be noted that, in evaluating the mean and variance of (12), instead of using a uniform mixing weight $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]$ for all coefficients, the state probability matrix $\mathbf{V}$ is utilized, where signal coefficients are assigned with non-uniform weights. More specifically, in (12), $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]$ is replaced with $[V_{n1}, \ldots, V_{nK}]$ for signal coefficient x_n, where $n \in [1, \ldots, N]$. Therefore, $p_{nk} = C_n \frac{V_{nk}}{Z_{nk}}$, $C_n = (\sum_k \frac{V_{nk}}{Z_{nk}})^{-1}$, and Z_{nk} can be calculated in (15).

B. First Order Approximation by Chain Rule and Matrix Operations

The above message updating strategies (6), (7) and (21) enable an approximate MMSE solution by tracking $\mathcal{O}(MN)$ messages. To further simplify the belief propagation, we adopt a first order approximation strategy [29], where a variable node x_n sends a uniform message to all check nodes $\underline{y} = [y_1, \ldots, y_M]$. Similarly, a check node y_m sends a uniform message back to all variable nodes $\underline{x} = [x_1, \ldots, x_N]$, after which only $\mathcal{O}(N)$ messages are needed to be updated in each belief propagation iteration.

It should be noted that the first order approximate strategy involves taking the derivatives of (22) with respect to κ_{nm}. As a_{nm} is involved in equations, taking the derivative directly on (22) as [7] is complicated and intractable for varying number of mixture density components K. Therefore, we apply the *Chain Rule*, where the derivative is obtained by decomposing (22) into simpler constituent functions, the derivatives of which are then evaluated, and eventually chained together to form the target derivative.

To this end, the following update rules (24) to (28) are derived,

$$a_{x_n}^{(i)} = \sum_{m=1}^{M} A_{mn} \mu_{y_m}^{(i)} + \mu_{x_n}^{(i)}, \tag{24}$$

$$\mu_{x_n}^{(i+1)} = \mathbb{F}_n(a_{x_n}^{(i)}, b^{2\,(i)}) = \sum_{k=1}^{K} p_{nk} E_{nk}^{(i)}, \tag{25}$$

TABLE I
MESSAGE PASSING PARAMETERS

$$\frac{\mathrm{d}\mathbb{F}_n}{\mathrm{d}a_{x_n}} = [\sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}} E_{nk}] \frac{\mathrm{d}C_n}{\mathrm{d}a_{x_n}} + C_n \sum_{k=1}^{K} V_{nk} \frac{\mathrm{d}\,(E_{nk}/Z_{nk})}{\mathrm{d}a_{x_n}}, \quad (I.1)$$

$$\frac{\mathrm{d}C_n}{\mathrm{d}a_{x_n}} = C_n^2 \sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}^2} \frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}}, \quad (I.2)$$

$$\frac{\mathrm{d}\,(E_{nk}/Z_{nk})}{\mathrm{d}a_{x_n}} = \frac{1}{Z_{nk}^2} \left(Z_{nk} \frac{\mathrm{d}E_{nk}}{\mathrm{d}a_{x_n}} - E_{nk} \frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}} \right), \quad (I.3)$$

$$\frac{\mathrm{d}E_{nk}}{\mathrm{d}a_{x_n}} = \frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}} + \zeta_{nk} \frac{\mathrm{d}\,(\phi(\eta_{nk})/\Phi(\eta_{nk}))}{\mathrm{d}a_{x_n}}, \quad (I.4)$$

$$\frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}} = \frac{\omega_k^2}{b^2 + \omega_k^2}, \quad (I.5)$$

$$\frac{\mathrm{d}\,(\phi(\eta_{nk})/\Phi(\eta_{nk}))}{\mathrm{d}a_{x_n}} = \frac{\mathrm{d}\phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \Phi^{-1}(\eta_{nk}) - \frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \frac{\phi(\eta_{nk})}{\Phi^2(\eta_{nk})}, \quad (I.6)$$

$$\frac{\mathrm{d}\phi(\eta_{nk})}{\mathrm{d}a_{x_n}} = -\eta_{nk} \phi(\eta_{nk}) \frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}}, \quad (I.7)$$

$$\frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} = \phi(\eta_{nk}) \frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}}, \quad (I.8)$$

$$\frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}} = \frac{h_k}{\sqrt{1 + h_k^2 \sigma_{nk}^2}} \frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}}, \quad (I.9)$$

$$\delta_{nk} = \frac{a_{x_n} - \xi_k}{\sqrt{b^2 + \omega_k^2}}, \quad (I.10)$$

$$\tau_{nk} = -\frac{1}{2} \sqrt{b^2 + \omega_k^2}\, (\phi(\delta_{nk}) \Phi(\eta_{nk}))^{-2}, \quad (I.11)$$

$$\frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}} = \tau_{nk} \left(\frac{\mathrm{d}\phi(\delta_{nk})}{\mathrm{d}a_{x_n}} \Phi(\eta_{nk}) + \frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \phi(\delta_{nk}) \right), \quad (I.12)$$

$$\frac{\mathrm{d}\phi(\delta_{nk})}{\mathrm{d}a_{x_n}} = -\frac{a_{x_n} - \xi_k}{b^2 + \omega_k^2} \phi(\delta_{nk}) \quad (I.13)$$

$$\begin{aligned}\sigma^{2\,(i+1)}_{x_n} &= \mathbb{G}_n(a_{x_n}^{(i)}, b^{2\,(i)}) \\ &= \sum_{k=1}^{K} p_{nk}[(E_{nk}^{(i)})^2 + Var_{nk}^{(i)}] - \left(\sum_{k=1}^{K} p_{nk} E_{nk}^{(i)}\right)^2,\end{aligned} \tag{26}$$

$$\mu_{y_m}^{(i+1)} = y_m - \sum_{n=1}^{N} A_{mn} \mu_{x_n}^{(i)} + \frac{\mu_{y_m}^{(i)}}{M} \sum_{n=1}^{N} \mathbb{F}'_n(a_{x_n}^{(i)}, b^{2\,(i)}), \tag{27}$$

$$b^{2\,(i+1)} = \hat{\sigma}_e^2 + \frac{1}{M} \sum_{n=1}^{N} \sigma^{2\,(i+1)}_{x_n}, \tag{28}$$

where $\mathbb{F}'_n \triangleq \frac{\mathrm{d}\mathbb{F}_n}{\mathrm{d}a_{x_n}}$ and related parameters are calculated as Table I, with iteration i being omitted for simplicity.

At implementation, $\mu_{y_m}^{(1)}$ in (24) is initialized at y_m for $m \in [1, \ldots, M]$, and $\mu_{x_n}^{(1)}$ is set to 0 for $n \in [1, \ldots, N]$. Besides, b^2 in (25) to (27) is initialized at 10^4 for robustness. Additionally, a maximum iteration of 100 is set for the approximate message passing module, and the convergence criteria is set to $\|\underline{\hat{\mu}}^{i+1} - \underline{\hat{\mu}}^{i}\|_2 \leq 10^{-8}$, where $\underline{\hat{\mu}}^{(i)} = [\mu_{x_1}^{(i)}, \ldots, \mu_{x_N}^{(i)}]$.

IV. Parameter Estimation: An Expectation-Maximization Approach

In this section, given the current reconstruction of the signal $\underline{\hat{x}}^i_{\text{MMSE}}$ from the approximate message passing module, a novel

Expectation-Maximization based algorithm is designed to learn the underlying parameter set $\mathbf{\Theta}$ that specifying the mixture.

A. Learning the Parameters

In our technique, the mixture density $\mathbf{\Theta}$ is obtained by seeking a MLE solution, $\widehat{\mathbf{\Theta}}^i_{\text{EM}} = \arg\max p\,(\underline{X} = \underline{\hat{x}}^i_{\text{MMSE}}|\mathbf{\Theta})$, using an Expectation-Maximization-based method.

For the ease of derivation in estimating the density parameters, it is assumed that signal coefficients are jointly independent. Therefore, the log-likelihood function can be written as,

$$\ln p(\underline{\hat{x}}|\underline{\lambda}, \mathbf{\Theta}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k) \right\} + \pi \left(\sum_{k=1}^{K} \lambda_k - 1 \right), \tag{29}$$

where the last term comes from the constraint $\sum_{k=1}^{K} \lambda_k = 1$, and π is a *Lagrange* multiplier.

Taking the derivative of (29) with respect to the mixing weight λ_k, and set it to 0, the following is derived,

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \mathbf{\Theta})}{\mathrm{d}\lambda_k} = \sum_{n=1}^{N} \frac{SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)}{\sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)} + \pi = 0. \tag{30}$$

Meanwhile, let

$$\gamma_{nk} = \frac{\lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)}{\sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)} \tag{31}$$

be the probability[3] of density component k on signal coefficient x_n. Given the above, and multiplying λ_k with (30), it is derived that,

$$\pi = -N, \quad \text{and} \tag{32}$$

$$\hat{\lambda}_k = \frac{\sum_{n=1}^{N} \gamma_{nk}}{N}, \tag{33}$$

where (32) holds due to fact $\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_{nk} = N$, and $\sum_{k=1}^{K} \lambda_k = 1$.

Besides, denote $\psi_{nk} = \phi(\alpha_k \frac{\hat{x}_n - \xi_k}{\omega_k})/\Phi(\alpha_k \frac{\hat{x}_n - \xi_k}{\omega_k})$, and ξ_k can then be updated by taking the derivative of (29) with respect to ξ_k, and setting it to 0,

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \Theta)}{\mathrm{d}\xi_k} = \sum_{n=1}^{N} \gamma_{nk} \left[\frac{\hat{x}_n - \xi_k}{\omega_k^2} - \frac{\alpha_k}{\omega_k} \psi_{nk} \right] = 0. \tag{34}$$

Similarly, taking the derivative of (29) with respect to ω_k gives

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \Theta)}{\mathrm{d}\omega_k} = \sum_{n=1}^{N} \frac{\gamma_{nk}}{\omega_k^3} [(\hat{x}_n - \xi_k)^2 - \omega_k^2 - \omega_k \alpha_k (\hat{x}_n - \xi_k) \psi_{nk}], \tag{35}$$

and ω_k is updated as

$$\omega_k^2 \sum_{n=1}^{N} \gamma_{nk} + \omega_k \alpha_k \sum_{n=1}^{N} \gamma_{nk} \psi_{nk} (\hat{x}_n - \xi_k) - \sum_{n=1}^{N} \gamma_{nk} (\hat{x}_n - \xi_k)^2 = 0. \tag{36}$$

Additionally, α_k can be updated by solving

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \Theta)}{\mathrm{d}\alpha_k} = \sum_{n=1}^{N} \gamma_{nk} \psi_{nk} \frac{(\hat{x}_n - \xi_k)}{\omega_k} = 0. \tag{37}$$

Therefore, (31), (34), (36)(a)nd (37) complete one iteration of the Expectation-Maximization update for γ_{nk}, ξ_k, ω_k, and α_k, where $k \in [1, \ldots, K]$, and $n \in [1, \ldots, N]$.

To summarize, our proposed Expectation-Maximization module starts with an initialization $\Theta^{(0)}$ and $\underline{\lambda}^{(0)} = [\lambda_1, \ldots, \lambda_K]$, and alternates between the following Expectation and Maximization steps,

1) *Expectation step:* Given the current mixture parameters $\mathbf{\Theta}^{(i)}$, evaluate the soft responsibility γ_{nk} for $k \in [1, \ldots, K]$, and $n \in [1, \ldots, N]$.
2) *Maximization step:* With updated soft responsibility, for $k \in [1, \ldots, K]$, re-estimate ξ_k, ω_k, and α_k using (34), (36), and (37) respectively.

where as in [31], [32], parameters are updated sequentially in our proposed method.

It should be pointed out that the learning rules (34), (36), and (37) for ξ_k, ω_k, and α_k are not in closed forms, thus the solutions cannot be calculated explicitly. In this case, one can take advantage of *root-finding* routines, including Golden Section, Newton's method, or Secant's Method [33], to solve for the solution.

B. Approximate ψ_{nk} Using Piecewise Functions

It is worth noticing that the learning rules of ξ_k (34), ω_k (36), and α_k (37) involve evaluating the *inverse mills ratio* [34], $\psi(t) = \frac{\phi(t)}{\Phi(t)}$, where $t = \alpha_k \frac{x_n - \xi_k}{\omega_k}$, for $k \in [1, \ldots, K]$, and $n \in [1, \ldots, N]$.

Since $\Phi(t) \to 0$ as $t \to -\infty$, the *inverse mills ratio* $\psi(t)$ is evaluated as *Not a Number* (NaN) when the operand goes to extremes, which prevents the *Expectation-Maximization* and *root finding* procedure from updating properly. As a motivating example, $\psi(t)$ is evaluated as *NaN* at $t = -40$, which will cause the root finding procedure terminate before convergence, and thus the correct solution cannot be found.

Given the fact $\psi(t)$ is not an elementary function,[4] our strategy is to substitute it with an approximate that allows for reliable and efficient evaluation for all real numbers $t \in \mathbb{R}$.

[3] Also known as soft responsibility in [30].

[4] $\psi(t) = \phi(t)/\Phi(t)$ is not elementary because the denominator $\Phi(t)$ is not elementary. As [7], evaluating $\psi(t) = \phi(t)/\Phi(t)$ is more than 10 times slower than scaler operations.

Inspecting the limit of $\phi(t)/\Phi(t)$ as $t \to -\infty$, and recall the $L'Hospital's$ rule [35], the following is derived,

$$\lim_{t\to-\infty} \frac{\phi(t)/\Phi(t)}{t} = \lim_{t\to-\infty} \frac{(\phi(t))'}{(t\Phi(t))'} \tag{38}$$

$$= \lim_{t\to-\infty} \frac{(\phi(t))''}{(t\Phi(t))''} \tag{39}$$

$$= \lim_{t\to-\infty} \frac{(t^2-1)\exp(-t^2/2)}{(2-t^2)\exp(-t^2/2)} = -1, \tag{40}$$

where (38) holds due to

$$\lim_{t\to-\infty} t\,\Phi(t) = \lim_{t\to-\infty} \frac{\Phi(t)}{1/t} = \lim_{t\to-\infty} \frac{-t^2}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) = 0, \tag{41}$$

and (39) holds due to

$$\lim_{t\to-\infty} (t\Phi(t))' = \lim_{t\to-\infty} (\Phi(t) + t\phi(t))$$

$$= \lim_{t\to-\infty} \frac{t}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right) = 0. \tag{42}$$

Meanwhile, taking the limit of $\psi(t)$ as $t \to +\infty$ gives,

$$\lim_{t\to+\infty} \frac{\phi(t)}{\Phi(t)} = \frac{\phi(t)}{1} = \phi(t). \tag{43}$$

The above limits suggest that $\psi(t)$ is asymptotically equivalent to $-t$, and $\phi(t)$, in the limit of $t \to -\infty$, and $t \to +\infty$, respectively. Therefore, a plausible approximate of $\psi(t)$ can be formed by joining an affine function, and a normal *pdf* function. To be more specific, it is intended to approximate $\psi(t)$ by $\widehat{\psi}(t)$ as,

$$\widehat{\psi}(t) = \begin{cases} a_1 t + a_2, & \text{if } t \le \Delta \\ c_0 \phi(\frac{t-\mu_0}{\sigma_0}), & \text{if } t > \Delta \end{cases} \tag{44}$$

where Δ is the boundary dividing the domain, a_1 and a_2 are the parameters defining the affine function, and c_0, μ_0, σ_0 are the corresponding parameters specifying the scaled normal *pdf* function.

We adopt a numerical approach, where the goal is set to solve for the approximate $\widehat{\psi}(t)$ by fitting (44) to the samples of $\psi(t) = \phi(t)/\Phi(t)$. Since the approximate (44) is not piecewise linear, finding the optimal parameters $(\Delta, a_1, a_2, c_0, \mu_0, \sigma_0)$ is intractable [36]. To this end, an effective *k-means* [30](b)ased greedy algorithm is designed in *Algorithm 1* to find the parameters of (44).

Algorithm 1 starts with a *pre-partition* step, and is followed by a loop that alternates between *piecewise fitting* and *re-partition* steps. In *pre-partition*, a set of evenly spaced sampling points $\underline{\delta} = [\delta_1, \ldots, \delta_q]$ are drawn from the interval $[\delta_-, \delta_+]$, with a step size ϵ. Subsequently, $\underline{\delta}$ is split at the boundary Δ into two vectors $\underline{\delta}_l$ and $\underline{\delta}_u$, where $[\underline{\delta}_l, \underline{\delta}_u] = \underline{\delta}$, and $v \le \Delta < w$ holds for $v \in \underline{\delta}_l$, $w \in \underline{\delta}_u$. Additionally, applying $\psi(t)$ to elements of $\underline{\delta}_l$ and $\underline{\delta}_l$, leads to the regressands $\underline{\psi}_l^{(1)}$ and $\underline{\psi}_u^{(1)}$, respectively.

To find the parameters of the approximate, at iteration i, $\underline{\psi}_l^{(i)}$ and $\underline{\psi}_u^{(i)}$ are fitted by the affine function and normal function (44), respectively. In Matlab, the *least square error* fit of (44) can be obtained by calling *polyfit* and *fit* functions, leading to $\hat{a}_1^{(i)} t + \hat{a}_2^{(i)}$, and $\hat{c}_0{}^{(i)} \phi(\frac{t-\hat{\mu}_0{}^{(i)}}{\hat{\sigma}_0{}^{(i)}})$, correspondingly.

Moreover, the intersection of two fitted functions can be found by solving for t^* of the following,

$$\hat{a}_1^{(i)} t^* + \hat{a}_2^{(i)} = \hat{c}_0{}^{(i)} \phi\left(\frac{t^* - \hat{\mu}_0{}^{(i)}}{\hat{\sigma}_0{}^{(i)}}\right). \tag{45}$$

The above completes one iteration of the *piecewise fitting* step. At iteration $i+1$, the data is re-partitioned by setting the boundary to the intersection of two fitted functions, i.e., $\Delta^{(i+1)} = t^*$, and the loop continues until the convergence of the boundary.

The fitted results utilizing *Algorithm 1* are shown in Fig. 4, where for numerical stability and efficiency, the interval $[\delta_l, \delta_u]$ is fixed to a limited range with $\delta_l = -30$, $\delta_u = 30$, and the sampling step is set to $\epsilon = 10^{-4}$.

Algorithm 1: Approximating $\psi(t) = \phi(t)/\Phi(t)$ by a Piecewise Function

Initialize: $\Delta^{(0)} = 10^3$, $\Delta^{(1)} = -2$, $\epsilon = 10^{-4}$, tol $= 10^{-8}$, $\delta_- = -30$, $\delta_+ = 30$, $I_{\max} = 100$, and $i = 1$

Algorithm:

Pre-partition:

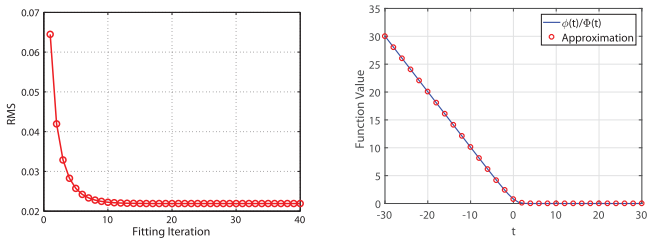
1) Build the sampling vector $\underline{\delta} = [\delta_1, \ldots, \delta_q]$ by drawing samples evenly from the interval $[\delta_-, \delta_+]$, with a step ϵ
2) Split $\underline{\delta}$ as $\underline{\delta}_l$ and $\underline{\delta}_u$ at the boundary Δ^1, such that $[\underline{\delta}_l, \underline{\delta}_u] = \underline{\delta}$, and $v \le \Delta^1 < w$ holds for $v \in \underline{\delta}_l$, $w \in \underline{\delta}_u$.
3) Build the regressands vectors $\underline{\psi}_l^{(1)}$ and $\underline{\psi}_u^{(1)}$ by applying $\psi(t)$ to $t \in \underline{\delta}_l$, and $t \in \underline{\delta}_u$, respectively.

while $i \le I_{\max}$ and $|\Delta^{(i)} - \Delta^{(i-1)}| \le$ tol, **do**

4) Fit affine function $a_1 t + a_2$ to $\underline{\psi}_l{}^{(i)}$,
$$[\hat{a}_1^{(i)}, \hat{a}_2^{(i)}] = \text{fit}(\underline{\psi}_l{}^{(i)})$$
5) Fit scaled normal *pdf* function $c_0\phi(\frac{t-\mu_0}{\sigma_0})$ to $\underline{\psi}_u{}^{(i)}$,
$$[\hat{c}_0{}^{(i)}, \hat{\mu}_0{}^{(i)}, \hat{\sigma}_0{}^{(i)}] = \text{fit}(\underline{\psi}_u{}^{(i)})$$
6) Find the intersection t^* of two fitted functions by solving,
$$\hat{a}_1^{(i)} t^* + \hat{a}_2^{(i)} = \hat{c}_0{}^{(i)} \phi\left(\frac{t^* - \hat{\mu}_0{}^{(i)}}{\hat{\sigma}_0{}^{(i)}}\right),$$
and update the boundary $\Delta^{(i+1)} = t^*$
7) Update $\underline{\psi}_l^{(i+1)}$ and $\underline{\psi}_u^{(i+1)}$ as of the steps in Pre-partition using the boundary $\Delta^{(i+1)}$
8) $i = i + 1$

end while

Return: $\Delta = \Delta^{(i)}$, $a_1 = \hat{a}_1^{(i)}$, $a_2 = \hat{a}_2^{(i)}$, $c_0 = \hat{c}_0^{(i)}$, $\mu_0 = \hat{\mu}_0^{(i)}$, and $\sigma_0 = \hat{\sigma}_0^{(i)}$.



(a) (b)

Fig. 4. Fit piecewise function to $\phi(t)/\Phi(t)$. (a) Root Mean Square (RMS) Errors of Fit. (b) Comparison of $\psi(t) = \phi(t)/\Phi(t)$ and its piecewise approximate $a_1 t + a_2$ for $t \leq \Delta$, and normal function $c_0 \phi(\frac{t-\mu_0}{\sigma_0})$ for $t > \Delta$, where $\Delta = -3.1727, a_1 = -0.994, a_2 = 0.1795, c_0 = 8.944, \mu_0 = -4.0153$, and $\sigma_0 = 2.2836$.

As can been seen in Fig. 4(a), the *Root Mean Square* (RMS) of the fit error[5] gradually decreases as the iteration increases, and eventually converges to RMS $= 0.022$, where the parameters are found to be $\Delta = -3.1727, a_1 = -0.994, a_2 = 0.1795, c_0 = 8.944, \mu_0 = -4.0153$, and $\sigma_0 = 2.2836$. Moreover, as can be seen in Fig. 4(b), the approximate (44) resembles $\psi(t) = \phi(t)/\Phi(t)$ quite decently.

C. Initialization Strategy

It is worth noticing that, as *Expectation-Maximization* only finds local optimums, a good initialization strategy is critical in building an effective parameter estimation procedure. In our work, given the number of mixture components K, the parameters are initialized by matching the moments of mixture component.

Specifically, the coefficients of current estimate $\underline{\hat{x}}$ are divided into K groups $\underline{\hat{x}} = [\underline{\hat{x}}_1, \ldots, \underline{\hat{x}}_K]$ by utilizing *k-means* algorithm [30].

Additionally, given the K clusters, the parameters for each density component are initialized in a way where sample mean, variance, and skewness match the population mean, variance, and skewness, respectively. Concretely, denote m_k as sample mean, ${v_k}^2$ as sample variance, and g_k as sample skewness, respectively. Then the location, scale, and shape parameters of skew normal density component k are initialized at ξ_k, ω_k, and α_k by solving,

$$m_k = \xi_k + \omega_k \frac{\alpha_k}{\sqrt{\pi(1+\alpha_k^2)/2}}, \tag{46}$$

$${v_k}^2 = \omega_k^2 \left(1 - \frac{2\alpha_k^2}{\pi(1+\alpha_k^2)}\right), \tag{47}$$

$$\left|\frac{\alpha_k}{\sqrt{1+\alpha_k^2}}\right| = \left(\frac{\pi}{2} \frac{|g_k|^{\frac{2}{3}}}{|g_k|^{\frac{2}{3}} + ((4-\pi)/2)^{\frac{2}{3}}}\right)^{\frac{1}{2}}, \tag{48}$$

where the sample skewness g_k is capped to a maximum absolute value of 0.95 for numerical stability, and the sign of α_k is same as g_k.

[5]RMS of a vector $\underline{\epsilon} \in \mathbb{R}^n$ is defined as $\underline{\epsilon}_{\text{rms}} = \sqrt{\frac{1}{N}(\epsilon_1^2 + \epsilon_2^2 + \cdots \epsilon_n^2)}$.

D. Estimate the Number of Density Components K

Selection of the number of components K is fundamental for techniques utilizing mixture model, and a variety of methods have been proposed to develop effective way for estimating K. In our work where the mixture component is skew normal, a non-parametric method is developed, where the number of components is estimated based on the modality of the kernel density estimate.

Specifically, given the signal coefficients, $\underline{\hat{x}} = [\hat{x}_1, \ldots, \hat{x}_N]$, a kernel $U : \mathbb{R} \rightarrow \mathbb{R}_+$, is placed at sample point $t \in \mathbb{R}$, and each signal coefficient $\hat{x}_n \in \underline{\hat{x}}$ contributes a non-negative density mass $U(t - \hat{x}_n)$. Utilizing the Gaussian kernel $U(t) = \phi(t)$, the density at sample point $t \in \underline{t}$, can be estimated by summing up the normalized contributions from all coefficients as,

$$\hat{f}(t) = \frac{1}{NW} \sum_{n=1}^{N} \phi\left(\frac{t - \hat{x}_n}{W}\right), \tag{49}$$

where $\underline{t} = [t_1, \ldots, t_L]$ is a vector of $L = 200$ evenly spaced sampling points drawn in the range of $\underline{\hat{x}}$, and W is the bandwidth that controls the spread of the density mass, and ultimately the smoothness of the density estimate.

It should be noted that the kernel density estimate found by (49) is highly sensitive to the choice of bandwidth W, where a large value leads to an *over-smoothed* estimate that under-fits the real density, and a small value makes the estimate *under-smoothed* and over-fits the real one. Therefore, a proper value of W is a good balance of *under-smoothing* and *over-smoothing*, where a well-behaved W is generally set manually by cross validation procedures.

In our work, the problem is tackled by a robust *two-stage* procedure. In the first place, the kernel density is estimated as (49), where the bandwidth is set to $W = 0.05$ to pick up the local variability of the density. Subsequently, a Gaussian weighted moving average filter is followed as the second stage to capture the overall modality of the underlying density, i.e.,

$$\hat{f}_g(t) = \sum_{j=1}^{W_f} \hat{f}(t - j + 1) V(j), \tag{50}$$

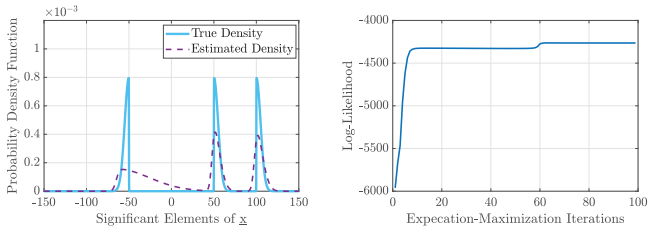
where $V(i) = \exp(\frac{-i^2}{2{\sigma_f}^2})$ is the Gaussian smoothing kernel,[6] with window size $W_f = 10$, and standard deviation $\sigma_f = 0.2 \times W_f = 2$. It is found out that although a good choice of W, W_f and σ_f are problem dependent, the above settings work decently in practice.

Given the above, the number of components K is estimated by counting the number of modes, i.e., $\hat{f}_g(i-1) < \hat{f}_g(i) < \hat{f}_g(i+1)$ for $i \in [1, \ldots, L]$. In Matlab, this can be obtained by calling the function *findpeaks*.

E. Evaluations of Parameter Estimation

Fig. 5 is a demonstration of the proposed *Expectation-Maximization* based mixture density estimation. To test the effectiveness of the module, a signal $\underline{x}$ is generated by drawing

[6]In practice, the kernel is normalized to $\sum_j V(j) = 1$, and the length of filtered output is same as the input.



(a) (b)

Fig. 5. Expectation-maximization mixture density estimate. (a) True and estimated mixture density of the significant coefficients. (b) Log-likelihood evaluated at expectation maximization iterations.

TABLE II
TRUE AND ESTIMATED PARAMETERS

	Density Parameters				Weight
$\underline{\theta}_1$	[$\xi_1 = 0$,	$\omega_1 = 1$,	$\alpha_1 = 0$	]	$\lambda_1 = 0.7$
$\hat{\underline{\theta}}_1$	[$\hat{\xi}_1 = 0$,	$\hat{\omega}_1 = 0.19$,	$\hat{\alpha}_1 = 0$	]	$\hat{\lambda}_1 = 0.65$
$\underline{\theta}_2$	[$\xi_2 = -50$,	$\omega_2 = 5$,	$\alpha_2 = -50$	]	$\lambda_2 = 0.1$
$\hat{\underline{\theta}}_2$	[$\hat{\xi}_2 = -65$,	$\hat{\omega}_2 = 39.05$,	$\hat{\alpha}_2 = 12.73$	]	$\hat{\lambda}_2 = 0.15$
$\underline{\theta}_3$	[$\xi_3 = 100$,	$\omega_3 = 5$,	$\alpha_3 = 50$	]	$\lambda_3 = 0.1$
$\hat{\underline{\theta}}_3$	[$\hat{\xi}_3 = 97.85$,	$\hat{\omega}_3 = 8.38$,	$\hat{\alpha}_3 = 2.96$	]	$\hat{\lambda}_3 = 0.1$
$\underline{\theta}_4$	[$\xi_4 = 50$,	$\omega_4 = 5$,	$\alpha_4 = 50$	]	$\lambda_4 = 0.1$
$\hat{\underline{\theta}}_4$	[$\hat{\xi}_4 = 47.90$,	$\hat{\omega}_4 = 7.92$,	$\hat{\alpha}_4 = 3.02$	]	$\hat{\lambda}_4 = 0.1$

$N = 2000$ random samples from a mixture of $K = 4$ skew normal density components, with the parameters being shown in Table II. Specifically, the insignificant coefficients of $\underline{x}$ are generated from skew normal density with parameter $\underline{\theta}_1$. Besides, the significant coefficients are generated from $\underline{\theta}_2$, $\underline{\theta}_3$, and $\underline{\theta}_4$. The mixing weights are set to $\lambda_1 = 0.7$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$, and $\lambda_4 = 0.1$, respectively. The density of significant coefficients is plotted in Fig. 5(a) with solid line.

The signal $\underline{x}$ is then sampled by (1), i.e., $\underline{y} = \mathbf{A}\underline{x} + \underline{e}$, with length of $\underline{y}$ being set to $M = 1650$. Meanwhile, the measurement white Gaussian noise $\underline{e}$ is added such that SNR $= 10\log_{10}(\frac{\|\mathbf{A}\underline{x}\|}{\|\underline{e}\|}) = 30$ dB, where $\|\underline{e}\| = \sum_{m=1}^{M} |e_m|^2$. Additionally, the signal reconstruction $\hat{\underline{x}}$ is obtained by employing the proposed signal inference module with an un-informative prior.

The proposed *Expectation-Maximization* module is applied to $\hat{\underline{x}}$ to estimate the mixture density, with the maximum iteration being set to 100. The log-likelihood of each iteration is tracked by evaluating (29), where convergence is reached when the consecutive difference of log-likelihood $\leq 10^{-6}$. Besides, the parameters found at each iteration are tracked, and the proposed module returns the solution with the maximum log-likelihood.

As can be seen in Fig. 5(b), the log-likelihood of the density estimate improves gradually as the iteration increases, and eventually converges with a gain of 1697.3. The estimated significant densities are plotted in Fig. 5(a) with dashed line. The true and estimated density parameters are compared at Table II. As can be seen, the proposed module recovers the number of mixture components as $\hat{K} = 4$ precisely. Besides, although deviated mildly in $\hat{\underline{\theta}}_2$, our technique faithfully recovers the overall modality and skewness of the signal.

V. States Estimation Using Belief Propagation and Potts Model

Given the reconstruction of the signal $\hat{\underline{x}}^i_{\text{MMSE}}$, and the estimated mixture density parameters $\hat{\mathbf{\Theta}}^i_{EM}$, in this section, we are aiming to promote the clustered property, and take inference of the underlying hidden states $\mathbf{S}$, by estimating the state probability matrix $\mathbf{V}$.

We approach the task by modelling the clustered property using the *Potts* model [37], where neighboring hidden state pairs are encouraged to be consistent through the regularization of the compatibility function. A belief propagation based technique is then employed to infer the hidden states and exploit clustered property by exchanging local beliefs.

A. Potts Model

In this work, a *K*-state *Potts* model is considered. Specifically, let $S_{i,j} \in [1, \ldots, K]$ be the hidden state variable of signal coefficient $X_{i,j}$, and $1 \leq i, j \leq d$. Besides, S_n and X_n correspond to $S_{i,j}$ and $X_{i,j}$ respectively, with the transform $n = (i-1) \times d + j$, and $1 \leq n \leq N$.

Borrowing the terminology from *Statistical Mechanics*, the energy of a hidden state configuration $\mathbf{S} = \mathbf{s} \in [1, \ldots, K]^N$ is defined as [38],

$$E(\mathbf{s}) = -\sum_{\langle u,v \rangle} J_0(s_u, s_v) - \sum_{n=1}^{N} H_0(s_n, \hat{x}_n), \tag{51}$$

where $J_0(s_u, s_v)$ is the *interaction* function that measures the consistency of neighboring hidden state pairs, $H_0(s_n, \hat{x}_n)$ is the *field* function that quantifies the coherence between estimated signal coefficients and the corresponding hidden states, and $\langle u, v \rangle$ denotes neighboring pairs.

Subsequently, denote $J(s_u, s_v) = \exp(J_0(s_u, s_v))$ as the *compatible* function, and let $H(s_n, \hat{x}_n) = \exp(H_0(s_n, \hat{x}_n))$ be the *evidence* function, the joint probability function of a hidden states $\mathbf{s}$ can be evaluated by *Boltzmann's law* as [38],

$$\begin{aligned} P(\mathbf{s}) &= \frac{1}{Z_p} \exp(-E(\mathbf{s})) \\ &= \frac{1}{Z_p} \prod_{\langle u,v \rangle} J(s_u, s_v) \prod_{n} H(s_n, \hat{x}_n), \end{aligned} \tag{52}$$

where Z_p is a normalization constant.

As can be seen from (51) and (52), *Potts* model can be configured by proper choice of *compatibility* and *evidence* functions,[7] such that *compatible* and *evident* hidden state configurations are preferred probabilistically over the *chaotic* counterparts.

B. Hidden State Inference by Belief Propagation

Given the *Potts* model, our goal is set to build appropriate *compatibility* and *evidence* functions, and then estimate the hidden state s_n for $n \in [1, \ldots, N]$ by computing the corresponding marginal probability from the joint probability (52). It should

[7] Or equivalently, *interaction* function and *field* function.

be noted that calculating the marginal probability involves summing over all other hidden state nodes, and unless N is very small, exact derivation is intractable in practice.

To this end, belief propagation is utilized to get an approximate estimate of the marginal probability by exchanging local beliefs.[8] Specifically, in the work, each density component of $\widehat{\mathbf{\Theta}}^i_{EM}$ is associated with a value of $s_n \in [1, \ldots, K]$, where the *evidence* $H(s_n, \hat{x}_n)$ is utilized to measure the responsibilities of mixture components on the specific signal coefficient. Therefore, the *evidence* function can be written as a *K-by-1* column vector,

$$\underline{H}_n = \underline{H}(s_n, \hat{x}_n) \cong [H_{n1}, \ldots, H_{nK}]^\intercal, \tag{53}$$

with $H_{nk} = \mathrm{SN}(\hat{x}_n | \hat{\xi}_k, \hat{\omega}_k, \hat{\alpha}_k)$.

Additionally, to promote clustered property, the compatibility function is defined in a way that neighboring pairs are encouraged to take identical hidden state. Therefore, following the vector representation of *evidence* function, the compatibility function is defined accordingly as a *K-by-K* state transition matrix [39],

$$\mathbf{J}^{(t)}(s_u, s_v) = \mathbf{J}^{(t)} = \tau^{(t)} \mathbf{I}_{K\times K} + \upsilon^{(t)}(\mathbf{1}_{K\times K} - \mathbf{I}_{K\times K}), \tag{54}$$

where t represents iteration, $\mathbf{I}_{K\times K}$ denotes identity matrix of size *K-by-K*, and $\mathbf{1}_{K\times K}$ represents matrix consisting of all ones. Besides, to promote compatible pairs, the compatibility function is made to be diagonally dominant by setting $\tau^{(t)} \gg \upsilon^{(t)}$, with the constraints $\tau^{(t)} + (K-1)\upsilon^{(t)} = 1$, and $0 \leq \tau^{(t)}, \upsilon^{(t)} \leq 1$.

Given the above, the state probability vector hidden state s_n can be calculated as the product of corresponding evidence and all incoming messages as [38], [39],

$$\hat{\underline{b}}_n^{(t)} \cong \underline{H}_n \bullet \prod_{j \in \mathrm{Neighbor}(n)}^{\bullet} \underline{m}_{jn}^{(t)}, \tag{55}$$

where $\underline{m}_{jn}^{(t)} \in \mathbb{R}^{K\times 1}$ denotes the message sending from s_j to its neighbor s_n, and can be evaluated as,

$$\underline{m}_{jn}^{(t)} \cong \mathbf{J}^{(t)} \left(\underline{H}_n \bullet \prod_{k \in \mathrm{Neighbor}(j)\setminus n}^{\bullet} \underline{m}_{kj}^{(t-1)} \right), \tag{56}$$

with $\bullet$ representing the *Hadamard* product [47] of vectors,[9] and $\mathrm{Neighbor}(j) \setminus n$ denoting the set of neighboring nodes s_j except s_n.

At implementation, the messages are initialized non-informatively at $\underline{m}_{ij}^{(0)} = [\frac{1}{K}, \ldots, \frac{1}{K}]^\intercal$ for all neighboring pairs $\langle i, j \rangle$. The messages are then propagated, and updated asynchronously [39], [40] by iteratively calling the message update rule (56) for 3 iterations. Besides, a first order neighborhood system is employed, where the hidden state $S_{u,v}$ statistically interacts with four adjacent neighbors, i.e., $S_{u,v+1}$, $S_{u,v-1}$, $S_{u+1,v}$, and $S_{u-1,v}$, for $1 \leq u, v \leq d$.

[8]Similar to the *message* in Section III, belief in this context encodes the marginal probability.

[9]*Hadamard* product of two vectors $\underline{a} = [a_1, a_2]^\intercal$ and $\underline{b} = [b_1, b_2]^\intercal$ gives another vector $\underline{a} \bullet \underline{b} = [a_1 b_1, a_2 b_2]^\intercal$.

Additionally, the hyper-parameters $\tau^{(t)}$ and $\upsilon^{(t)}$ are set based on the compatibility as,

$$\tau^{(t)} = \frac{r_s^{(t)}}{r_s^{(t)} + r_d^{(t)}}, \tag{57}$$

$$\upsilon^{(t)} = \frac{1}{K-1}\big(1 - \tau^{(t)}\big), \tag{58}$$

where $r_s^{(t)}$ and $r_d^{(t)}$ are updated with corresponding momentum and compatibility measure as,

$$r_s^{(t)} = r_s^{(t-1)} + \frac{\kappa^{(t)}}{\vartheta^{(t)} + \kappa^{(t)}}, \tag{59}$$

and

$$r_d^{(t)} = r_d^{(t-1)} + \frac{\vartheta^{(t)}}{\vartheta^{(t)} + \kappa^{(t)}}. \tag{60}$$

It should be noted that in the above, the compatibility measures $\kappa^{(t)}$ and $\vartheta^{(t)}$ are evaluated as the number of compatible pairs and incompatible pairs respectively, where at iteration t, a pair $\langle u, v \rangle$ are said to be compatible if they have identical dominant state, i.e., $\mathrm{argmax}\big(\hat{\underline{b}}_u^{(t)}\big) = \mathrm{argmax}\big(\hat{\underline{b}}_v^{(t)}\big)$, or incompatible otherwise.

VI. Complexity Analysis

Similar to other approximate message passing based techniques [7], [28], [29], our signal inference module is highly efficient. Concretely, the complexity of the module is dominated by two major operations. The first comes from evaluating (24), which when implemented by matrix, leads to the multiplication of a matrix of size $\mathbb{R}^{M\times N}$ with a vector of size $\mathbb{R}^{N\times 1}$. Therefore, a Floating Point Operations (FLOP) proportional to $\mathcal{O}(M(2N-1))$ is expected. The second rises from (25) and (26), which calls for the element-wise product of size $\mathbb{R}^{N\times K}$, leading to a FLOP of $\mathcal{O}(NK)$. As $K \ll M$ holds in practice, the overall FLOP of the approximate message passing module is $\mathcal{O}(T_s M(2N-1))$, where the maximum iteration is capped to $T_s = 100$.

Besides, the parameter estimation module involves finding the root of the function consisting of N terms for each of K density components. Considering the overhead [41] of root finding procedure,[10] and the fact that each density component has 3 parameters, the FLOP is expected to be $\mathcal{O}(15T_{em}KN)$ Expectation-Maximization module, where the maximum iteration is set to $T_{em} = 100$.

Additionally, the state estimation module enjoys great computation efficiency as well. Specifically, as (56) involves only element-wise product, a FLOP of $\mathcal{O}(4T_pKN)$ is expected for belief propagation, where the leading constant 4 comes from the size of neighborhood, and maximum iteration of $T_p = 3$ is set for the state estimation module.

Moreover, as discussed in Section II-B, our technique alternates between the aforementioned individual modules, and a maximum global iteration $I = 4$ is adopted. Therefore although

[10]A factor of $\log_2(32) = 5$ is anticipated for root finding procedure using *Newton's* method with a 32 digits precision representation.

TABLE III
FLOPS OF ALGORITHMS (VALUES ARE TRIMMED FOR VISIBILITY, AND LEADING CONSTANTS COME FROM THE DEFAULT MAXIMUM ITERATIONS OF CORRESPONDING ALGORITHMS)

Proposed	**CluSS**	**PCSBL**	**SPGL1**
$\mathcal{O}(800MN)$	$\mathcal{O}(N^2)$	$\mathcal{O}(400MN)$	$\mathcal{O}(10M^2N)$
SOMP	**BCS**	**EMGMAMP**	**SNAMP**
$\mathcal{O}(QMN)$	$\mathcal{O}(MN+100N)$	$\mathcal{O}(20MN)$	$\mathcal{O}(160MN)$

involving multiple modules, our proposed technique is highly efficient in exploring the salient features of the signals. As a rule of thumb, the time complexity of our proposed technique is estimated to be $\mathcal{O}(800MN)$ FLOP.

Table III shows the complexities of several sophisticated algorithms. To compare, CluSS calls for a FLOP of $\mathcal{O}(N^2)$ [17], whereas in PCSBL [48], assuming a default maximum iteration of 400, a FLOP of $\mathcal{O}(400\,MN)$ is expected. Additionally, as noted in [50], SOMP has a similar computational complexity as Orthogonal Matching Pursuit (*OMP*) [49], and thus a FLOP of $\mathcal{O}(QMN)$ is anticipated [49], with Q being the number of significant coefficients. Analyzing the structure of Bayesian Compressive Sensing (*BCS*) [44] reveals that its FLOP scales as $\mathcal{O}(MN+100N+\delta)$, where the overhead δ scales in the cube of a small fraction of N, which is negligible in practice. In Spectral Projected-Gradient ℓ_1 minimization (*SPGL1*) [43], the complexity is dominated by matrix-vector product, therefore a FLOP scales as $\mathcal{O}(10M^2N)$ is needed, where quadratic term of M comes from matrix-vector operations (MN) and maximum iteration $10M$. The FLOPs of Expectation-Maximization Gaussian Mixture Approximate Message Passing (*EMGAMAMP*) [42] and Skew-Normal Approximate Message Passing (SNAMP) are found to be $\mathcal{O}(20MN)$ and $\mathcal{O}(160MN)$, respectively.

VII. EXPERIMENTS

In this section, the performance of our proposed method is evaluated under a variety of numerical simulations. For each test, the signal $\underline{x}$ is sampled by (1), where the coefficients of the sampling matrix $\mathbf{A}$ are drawn from *i.i.d.* Gaussian ensemble, with the columns of $\mathbf{A}$ being normalized to unit ℓ_2 norm, i.e., $A=[\underline{A}_1{}^\intercal,\ldots,\underline{A}_N{}^\intercal]^\intercal$, and $\|\underline{A}_n\|_2=(\sum_{m=1}^{M}A_{mn}^2)^{\frac{1}{2}}=1$, for $1\le n\le N$.

At the reconstruction phase, the signal is estimated by the proposed technique that alternates among signal inference, mixture density estimate, and hidden state inference modules. The process is executed for a maximum of $i=4$ iterations, or till the convergence of reconstruction, i.e., $\|\hat{\underline{x}}^i-\hat{\underline{x}}^{i-1}\|_2^2/\|\hat{\underline{x}}^i\|_2^2 \le 10^{-4}$.

At iteration $i=1$, an un-informative setting is adopted, where the mixture is assumed to consist of $K=2$ normal density components, and the parameters are set to $\hat{\mathbf{\Theta}}^0=[\underline{\theta}_1,\underline{\theta}_2]$, where $\underline{\theta}_1=[\xi_1=0,\omega_1=0.5,\alpha_1=0]$, and $\underline{\theta}_2=[\xi_2=0,\omega_2=50,\alpha_2=0]$. Besides, the corresponding mixing weights are assumed to be $\lambda_1=0.8$, and $\lambda_2=0.2$. At iteration $i=1$, the state probability matrix is set to $\widehat{\mathbf{b}}^0=[\underline{b}_1,\ldots,\underline{b}_N]^\intercal$, with $\underline{b}_n=[\lambda_1,\lambda_2]^\intercal$, for $n\in[1,\ldots,N]$. The variance of measurement noise in (28) is initialized at $\hat{\sigma}_e^2=1$, and can be estimated based on residual as $\hat{\sigma}_e^2=\frac{1}{M}\|\underline{y}-\mathbf{A}*\hat{\underline{x}}\|_2^2$.

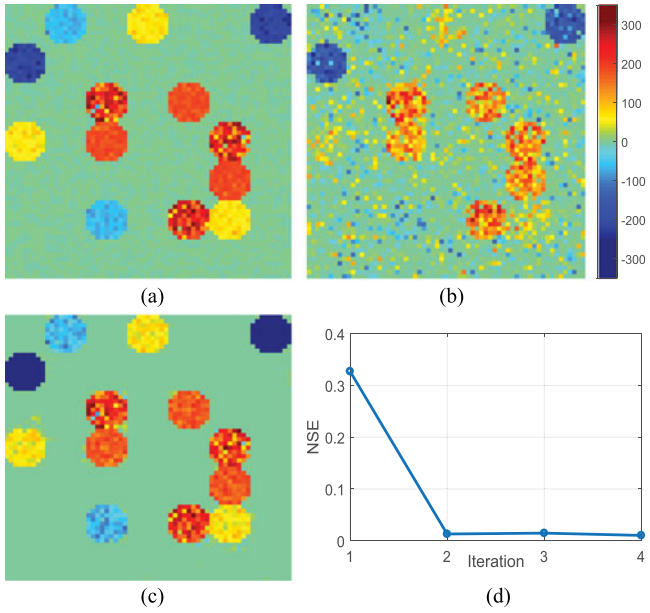


Fig. 6. Pictorial Demonstration. (a) Ground truth of the signal of size 63-by-63 that consists of $G_S=13$ significant clusters. (b) Reconstruction at iteration $i=1$, with $NMSE=8.24\times10^{-5}$. (c) Reconstruction at iteration $i=4$, with $NMSE=2.62\times10^{-6}$. (d) *NMSE* vs. iterations.

TABLE IV
MIXTURE DENSITY PARAMETERS

Density Parameters				
$\underline{\theta}_1$	$[\ \xi_1=0,$	$\omega_1=0.5,$	$\alpha_1=0$	$]$
$\underline{\theta}_2$	$[\ \xi_2=50,$	$\omega_2=20,$	$\alpha_2=5$	$]$
$\underline{\theta}_3$	$[\ \xi_3=-50,$	$\omega_3=20,$	$\alpha_3=-5$	$]$
$\underline{\theta}_4$	$[\ \xi_4=200,$	$\omega_4=20,$	$\alpha_4=-10$	$]$
$\underline{\theta}_5$	$[\ \xi_5=-200,$	$\omega_5=20,$	$\alpha_5=-10$	$]$
$\underline{\theta}_6$	$[\ \xi_6=300,$	$\omega_6=120,$	$\alpha_6=-10$	$]$

A. Pictorial Demonstration

As a demonstration, in this test, our proposed technique is examined by reconstructing an artificial signal $\mathbf{x}\in\mathbb{R}^{63\times63}$ shown in Fig. 6(a), with the length of the signal being $N=3969$. The coefficients are drawn from a mixture consisting of $K=6$ density components shown in Table IV, where without loss of generality, $\underline{\theta}_1$ denotes insignificant density component, and $\underline{\theta}_2$ to $\underline{\theta}_6$ represent significant density components.

As can be seen in Fig. 6(a), the signal $\underline{x}$ consists of $G_s=13$, *disk*-like significant clusters, with each cluster composing of 69 coefficients. In Table IV, the *Weight* of each density component is adjusted by the number of clusters, which are set to 3, 2, 3, 2, and 3, for $\underline{\theta}_2$, $\underline{\theta}_3$, $\underline{\theta}_4$, $\underline{\theta}_5$, and $\underline{\theta}_6$, respectively. The signal is sampled by (1), where the number of samples is set to $M=1794$, and the measurement is noisy with SNR $=35$ dB.

The signal is then reconstructed by our proposed technique, and Fig. 6(b) and 6(c) show the reconstruction obtained at 1st,

and 4th iteration, respectively. As can be seen in Fig. 6(b), the reconstruction of 1st iteration missed 5 clusters, and the signal estimate is corrupted by a large number of *salt-and-pepper* noises. After a few iterations, our proposed technique manages to recover all clusters, and as can be seen in Fig. 6(c), the reconstruction of the last iteration reliably resembles the ground truth of the signal.

The reconstruction error is tracked by evaluating NMSE $\triangleq \frac{1}{N}\|\hat{\underline{x}} - \underline{x}\|_2^2/\|\underline{x}\|_2^2$ at each iteration, and is plotted in Fig. 6 (d). As can be seen, our proposed technique faithfully reduces the reconstruction error, which eventually delivers NSE $= 0.0104$ at the last iteration.

B. Phase Transition

In the second test, the performance of our proposed algorithm is evaluated under the phase transition test. Concretely, the size of the signal is fixed to 54-by-54, with the length $N = 2916$. Besides, M/N is varied from 0.1 to 0.5, at 0.05 intervals. Additionally, for each value of M, the number of significant clusters G_s, is varied from 1 to $\lfloor \frac{M}{d} \rfloor$, at steps of 1. Similar to previous tests, the shape of cluster is *disk*, and each cluster consists 69 coefficients.[11] The signal coefficients are drawn from the density mixture shown in Table IV, where a maximum of 5 significant densities, i.e., $\underline{\theta}_2$ to $\underline{\theta}_6$, is considered. 200 independent trials are performed for each combination of M and G_s, and for each trial, the number of clusters corresponding to each significant density, is generated uniform randomly.

Our proposed method is compared with several sophisticated *structure-aware* methods, including SOMP [16], SRL1 [11], SPGL1 [43], and BCS [44]. Additionally, our proposed algorithm also compared to SNAMP [7] which is designed for asymmetrical sparse signals. It should be noted that, SOMP requires the prior knowledge of the number of significant coefficients. Therefore, for fairness, similar to the setting of our proposed technique, the sparsity in SOMP is set to 0.2.

Similar to [46] and [7], success rate is employed to measure the goodness of the methods, and a successful trial is defined as the one with NMSE $\leq 10^{-4}$. The results are summarized in Fig. 7, where Q/M vs M/N is depicted, and $Q = 69 \times G_s$ represents the number of significant coefficients. Similar to [46], the area under each curve represents the range at which the corresponding success rate $\geq 50\%$.

It can be seen in Fig. 7 that our proposed method gives competitive results in the phase transition tests. Specifically, our technique is most effective when $M/N > 0.3$. We believe this advantage comes from the fact that mixture estimation requires sizable significant coefficients to be efficient.

C. Noisy Reconstruction

In this test, our scheme is tested under noisy environments. Specifically, Gaussian random noise $\underline{e}$ is added to the measurements as in (1). Similar to *Phase Transition tests*, the size of signal is set to 54-by-54. The signal coefficients are drawn from the density mixture defined in Table IV. A total of $G_s = 15$ significant clusters are generated, with each significant density, i.e., $\underline{\theta}_2$ to $\underline{\theta}_6$, contributing 3 clusters.

[11] $\lfloor \frac{M}{d} \rfloor$ represents the largest integer $\leq \frac{M}{d}$.

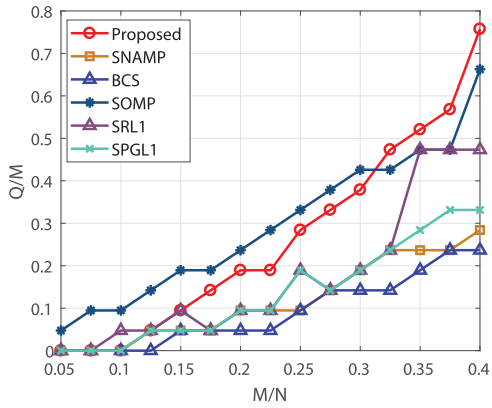


Fig. 7. Phase Transition tests. The size of significant cluster is set to $d = 69$, and the number of significant coefficients is $Q = 69 \times G_s$. M/N is varying from 0.1 to 0.4 at 0.05 intervals, and Q/M is varying by increasing G_s from 1 to $\lfloor \frac{M}{d} \rfloor$ at steps of 1.

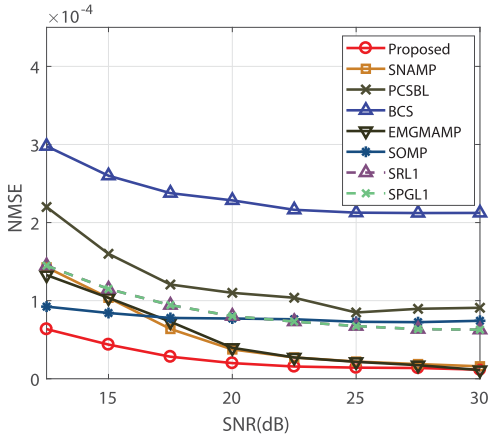


Fig. 8. NMSE vs. SNR.

Fig. 8 shows the reconstruction *NMSE* under noisy environments, where *SNR* is varied from 12.5 dB to 30 dB, at 2.5 dB intervals, and each data point is averaged over 200 independent trials. It can be seen from Fig. 8 that, our proposed technique CL-SNM-BP gives superior results under varying SNRs.

D. Runtime Tests

The time complexity of our proposed algorithm is evaluated by the *Runtime tests*. The size of signal is set to d-by-d, where d varies from 18 to 72, at steps of 9. The shape of significant clusters is *disk*, with each containing 69 coefficients. Besides, the number of significant clusters is fixed to $G_s = 2$, with one cluster drawing from $\underline{\theta}_3$, and the other sampling from $\underline{\theta}_4$ of Table IV. Additionally, the number of measurements is set to $M = 276$.

The experiments are performed on a desktop with hex core 3.2 GHz CPUs, and 16 GB of 1333 MHz memory. 20 independent trials are performed for each value of d, and the runtime of our proposed technique is compared to SNAMP, Sparse

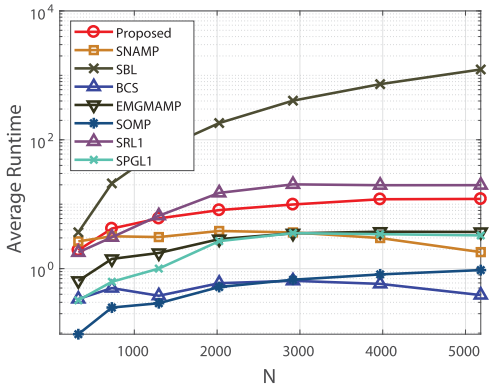


Fig. 9. Signal Length N vs. Average Runtime (in seconds).

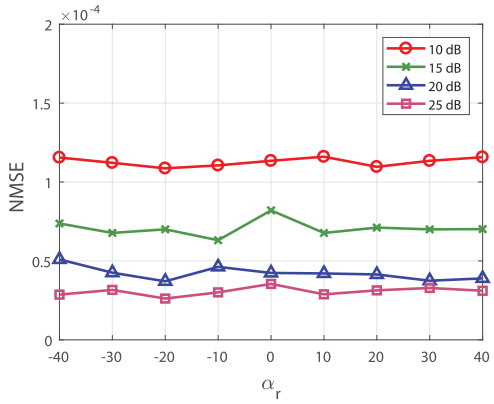


Fig. 10. NMSE vs. shape parameter α_r.

TABLE V
ROBUSTNESS TEST MIXTURE DENSITY PARAMETERS

Density Parameters				
$\underline{\theta}_1$	[$\xi_1 = 0,$	$\omega_1 = 0.5,$	$\alpha_1 = 0$	]
$\underline{\theta}_2$	[$\xi_2 = 200,$	$\omega_2 = 20,$	$\alpha_2 = \alpha_r$	]
$\underline{\theta}_3$	[$\xi_3 = -200,$	$\omega_3 = 20,$	$\alpha_3 = \alpha_r$	]

Bayesian Learning (*SBL*) [45], BCS, EMGMAMP, SOMP, SRL1, and SPGL1. Fig. 9 shows the average runtime of each method as the size of the signal N increases.

It can be seen that, as multiple modules are involved in our proposed technique, the runtime of our scheme is slightly longer than other approximate message passing relatives, i.e., , EMGMAMP, and SNAMP. Yet it should be pointed out that, our proposed algorithm scales decently with the increment of N. Specifically, reconstruction of the signal with $N = 324$ leads to an average runtime of 1.94 seconds, which is then scaled to 12.08 seconds when $N = 5184$.

E. Robustness Test

In this experiment, we are interested in analyzing the robustness of our scheme by feeding signals with density components of different levels of skewness. This is done by generating the signal coefficients from Table V, and varying shape parameters from -40 to 40, at steps of 10.

The size of the signals is 54-by-54, with $N = 2916$, and $M = 1449$. Besides, $G_s = 12$ significant clusters are generated, with each significant density contributing 6 *disk* clusters of size 69. Our proposed scheme is tested under noisy environments, where *SNRs* vary from 10 dB to 25 dB.

The results are summarized in Fig. 10, where each data point is averaged over 200 independent trials. It should be noted that, in Fig. 10, $\alpha_r = +40$ ($\alpha_r = -40$) represents approximately the positive (negative) half-normal density. On the other hand, $\alpha_r = 0$ resembles the normal density. As can be seen, in general, our proposed technique can adapt to different skewness, and provides robust and consistent reconstruction when the signal is generated from varying shape parameters α_r.

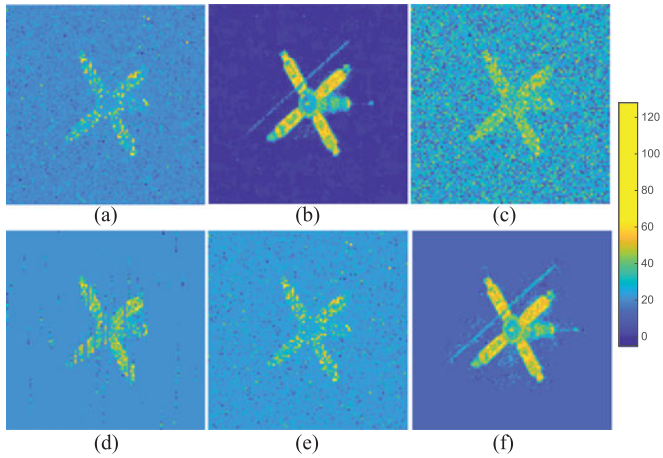


Fig. 11. Reconstruction of Satellite image. (a) the proposed algorithm at iteration $i = 1$ ($NMSE = 0.5978$), (b) the proposed algorithm at iteration $i = 4$ ($NMSE = 0.0149$), (c) *PCSBL* ($NMSE = 0.7842$), (d) *CluSS* ($NMSE = 0.4113$), (e) *EMGMAMP* ($NMSE = 0.5755$), and (f) *SOMP* ($NMSE = 0.0324$).

F. Satellite Image Recovery

In this test, our proposed technique is applied to the real-world *Satellite* image shown in Fig. 1(a), where the size of image is $N = 93 * 93 = 8694$, and the number of measurement is set to $M = 2000$. Our technique is compared with SOMP [16], EMGMAMP [42], SPGL1 [43], BCS [44], PCSBL [48], and SNAMP [7].

Fig. 11 shows the reconstructions of different methods under a noisy environment with SNR $= 25$ dB. It can be seen that, our proposed technique gives a competitive reconstruction under this test. It is also noteworthy that although the reconstruction at iteration $i = 1$ (Fig. 11(a)) is highly corruptive (NMSE $=$ 0.5978), our proposed technique manages to recover most of significant clusters at iteration $i = 4$ (Fig. 11(b)), and ends up with NMSE $= 0.0149$.

VIII. CONCLUSION

The present work studied the compressive sensing task of clustered sparse signals, where the magnitudes of each significant cluster are distributed asymmetrically *w.r.t* the cluster mean.

To capture the skewness feature, a finite skew normal density mixture is utilized to model the prior distribution of the signal. The clustered property is modelled by the *Potts* model. An effective algorithm, *CL-SNM-BP*, is developed to estimate the signal by alternating among exploiting the measurement, drawing inference of the finite skew normal mixture, and taking advantage of the clustered property. Experiments under a variety of settings show that our technique is effective in exploring both the skewness, and the clustered features of the signals.

It should be noted that, despite the numerical studies, rigorous theoretical analysis of the convergence behavior of our technique is worth further investigation in future research. Besides, as many of hyper-parameters are chosen empirically in the current work, it is therefore a meaningful extension to study the optimality of those parameters.

APPENDIX

Proof: Similar to the *Lemma 2* of [7], the moment generating function of X is derived as,

$$M_X(t) = Z \int \exp(tx)\mathcal{N}(x; a, b^2)\mathcal{SN}(x; \xi, \omega, \alpha)\mathrm{d}x \tag{61}$$

$$= \frac{\exp(t\mu + t^2\frac{\sigma^2}{2})}{\sqrt{2\pi\sigma^2}\Phi(\eta)} \int \exp\left(\frac{\left(x - (\mu + t\sigma^2)\right)^2}{-2\sigma^2}\right) \Phi\left(\alpha\frac{x-\xi}{\omega}\right) \mathrm{d}x \tag{62}$$

$$= \frac{\exp(t\mu + t^2\frac{\sigma^2}{2})}{\Phi(\eta)} \Phi\left(\frac{\kappa + h(\mu + t\sigma^2)}{\sqrt{1 + h^2\sigma^2}}\right) \tag{63}$$

where (62) holds by combining the exponential terms, and (63) holds due to *Lemma 1* of [7]. Subsequently, taking the derivative to the $M_X(t)$, and setting $t = 0$, the mean of X is found to be,

$$\mathrm{E}(X) = M_x'(0) = \mu + \frac{h\sigma^2}{\sqrt{1 + h^2\sigma^2}} \frac{\phi(\eta)}{\Phi(\eta)}. \tag{64}$$

Similarly, the variance of X is derived as,

$$\begin{aligned}\mathrm{Var}(X) &= M_x''(0) - \left(M_x'(0)\right)^2 \\ &= \mu^2 + \sigma^2 + \rho\zeta\frac{\phi(\eta)}{\Phi(\eta)} - \mathrm{E}^2(X),\end{aligned} \tag{65}$$

where $\zeta = \frac{h\sigma^2}{\sqrt{1+h^2\sigma^2}}$, and $\rho = \frac{2\mu + \mu h^2\sigma^2 - \kappa h\sigma^2}{1+h^2\sigma^2}$. ■

ACKNOWLEDGMENT

S. Wang would like to thank W. Xiao for her considerable help in proofreading the paper. The authors would like to thank the anonymous reviewers for their valuable inputs that have greatly improved the quality of the paper.

REFERENCES

[1] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Math.*, vol. 346, pp. 589–592, 2008.
[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[3] E. J. Candès, "Compressive sampling," in *Proc. Int. Congr. Math.*, 2006, vol. 3, pp. 1433–1452.
[4] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
[5] Z. Pan *et al.*, "Super-resolution based on compressive sensing and structural self-similarity for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4864–4876, Sep. 2013.
[6] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang, "Low-dose X-ray CT reconstruction via dictionary learning," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1682–1697, Sep. 2012.
[7] S. Wang and N. Rahnavard, "A framework for compressive sensing of asymmetric signals using normal and skew-normal mixture prior," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5062–5072, Dec. 2015.
[8] A. Talari and N. Rahnavard, "Cstorage: Distributed data storage in wireless sensor networks employing compressive sensing," *Ad Hoc Netw.*, vol. 37, pp. 475–485, 2016.
[9] B. Shahrasbi and N. Rahnavard, "Model-based nonuniform compressive sampling and recovery of natural images utilizing a wavelet-domain universal hidden Markov model," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 95–104, 2017.
[10] S. Pudlewski, A. Prasanna, and T. Melodia, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1060–1072, Jun. 2012.
[11] S. Wang, B. Shahrasbi, and N. Rahnavard, "SRL1: Structured reweighted l1 minimization for compressive sampling of videos," in *Proc. Int. Symp. Inf. Theory*, 2013, pp. 301–305.
[12] M. Nguyen, K. Teague, and N. Rahnavard, "CCS: Energy-efficient data collection in clustered wireless sensor networks utilizing block-wise compressive sensing," *Comput. Netw.*, vol. 106, pp. 171–185, 2016.
[13] S. Wang and N. Rahnavard, "Binary compressive sensing via sum of L-1 norm and L-infinity norm regularization," in *Proc. IEEE Mil. Commun. Conf.*, 2013, pp. 1616–1621.
[14] U. Nakarmi and N. Rahnavard, "BCS: Compressive sensing for binary sparse signals," in *Proc. IEEE Mil. Commun. Conf.*, 2012, pp. 1–5.
[15] [Online]. Available: http://www.nws.noaa.gov/asos/
[16] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *J. Mach. Learn. Res.*, vol. 12, pp. 3371–3412, 2011.
[17] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Process.*, vol. 92, no. 1, pp. 259–269, 2012.
[18] V. Cevher, M. F. Duarte, C. Hegde, and R. Baraniuk, "Sparse signal recovery using Markov random fields," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 257–264.
[19] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, Jun. 2016.
[20] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 2009–2015, Apr. 2013.
[21] T. Peleg, Y. C. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2286–2303, May 2012.
[22] M. E. Eltayeb, T. Y. Al-Naffouri, and H. R. Bahrami, "Compressive sensing for feedback reduction in MIMO broadcast channels," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3209–3222, Sep. 2014.
[23] [Online]. Available: http://www1.ncdc.noaa.gov/pub/download/asos/
[24] J. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4689–4703, Sep. 2014.
[25] A. Azzalini, "A class of distributions which includes the normal ones," *Scand. J. Statist.*, vol. 12, pp. 171–178, 1985.
[26] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012.
[27] B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model.* Cambridge, MA, USA: Harvard Univ. Press, 2014.
[28] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. IEEE Inf. Theory Workshop*, 2010, pp. 1–5.
[29] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: II. Analysis and validation," in *Proc. IEEE Inf. Theory Workshop*, 2010, pp. 1–5.
[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[31] T. I. Lin, J. C. Lee, and S. Y. Yen, "Finite mixture modelling using the skew normal distribution," *Statistica Sinica*, vol. 17, pp. 909–927, 2007.
[32] X. L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.
[33] S. S. Rao, *Engineering Optimization: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2009.
[34] J. P. Mills, "Table of the ratio: Area to bounding ordinate, for any portion of normal curve," *Biometrika*, pp. 395–400, 1926.
[35] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, vol. 55. New York, NY, USA: Dover, 1964.
[36] A. Magnani and S. P. Boyd, "Convex piecewise-linear fitting," *Optim. Eng.*, vol. 10, no. 1, pp. 1–17, 2009.
[37] R. B. Potts, "Some generalized order-disorder transformations," *Math. Proc. Cambridge Philosoph. Soc.*, vol. 48, no. 1, pp. 106–109, 1952.
[38] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring Artif. Intell. New Millennium*, vol. 8, pp. 236–239, 2003.
[39] Z. Yin and R. Collins, "Belief propagation in a 3D spatio-temporal MRF for moving object detection," in *Proc. IEEE. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[40] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 900–906.
[41] [Online]. Available: http://en.citizendium.org/wiki/Newton%27s_method
[42] J. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.
[43] E. V. D. Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.
[44] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
[45] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
[46] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosoph. Trans. Roy. Soc. London A, Math. Phys. Eng. Sci.*, vol. 367, no. 1906, pp. 4273–4293, 2009.
[47] C. Bocci, E. Carlini, and J. Kileel, "Hadamard products of linear spaces," *J. Algebra*, vol. 448, pp. 595–617, 2016.
[48] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, Jun. 2016.
[49] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[50] J. Huang, "Structured sparsity: Theorems, algorithms and applications," Ph.D. dissertation, Dept. Comput. Sci., Rutgers Univ., New Brunswick, NJ, USA, 2011.

Sheng Wang (S'15) received the B.S. degree in instrumentation and control engineering from the Hefei University of Technology, Hefei, China, in 2008, the M.S. degree in instrumentation and control engineering from Tianjin University, Tianjin, China, in 2010, and the Ph.D. degree from the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA, in 2017. His research interests include compressive sensing, statistical signal processing, and deep learning.

Nazanin Rahnavard (S'97–M'10) received the B.S. and M.S. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1999 and 2001, respectively. She then received the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. She is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL, USA. She has interest and expertise in a variety of research topics in the communications, networking, and signal processing areas. She is the recipient of the NSF CAREER award in 2011. She also received the 2007 Outstanding Research Award from the Center of Signal and Image Processing at Georgia Tech. She serves on the editorial board of the Elsevier *Journal on Computer Networks* (COMNET) and on the Technical Program Committee of several prestigious international conferences.