Compressive Sensing by Learning a Gaussian Mixture Model from Measurements

Jianbo Yang, *Member, IEEE*, Xuejun Liao, *Senior Member, IEEE*, Xin Yuan, *Member, IEEE*, Patrick Llull, David J. Brady, *Fellow, IEEE*, Guillermo Sapiro, *Fellow, IEEE*, and Lawrence Carin, *Fellow, IEEE*

***Abstract*—Compressive sensing of signals drawn from a Gaussian mixture model (GMM) admits closed-form minimum mean squared error (MMSE) reconstruction from incomplete linear measurements. An accurate GMM signal model is usually not available *a priori*, because it is difficult to obtain training signals that match the statistics of the signals being sensed. We propose to solve that problem by learning the signal model *in situ*, based directly on the compressive measurements of the signals, without resorting to other signals to train a model. A key feature of our method is that the signals being sensed are treated as random variables and are integrated out in the likelihood. We derive a maximum marginal likelihood estimator (MMLE), that maximizes the likelihood of the GMM of the underlying signals given only their linear compressive measurements. We extend the MMLE to a GMM with dominantly low-rank covariance matrices, to gain computational speedup. We report extensive experimental results on image inpainting, compressive sensing of high-speed video, and compressive hyperspectral imaging (the latter two based on real compressive cameras). The results demonstrate that the proposed methods outperform state-of-the-art methods by significant margins.**

***Index Terms*—Compressive sensing, Gaussian mixture model (GMM), mixture of factor analyzers (MFA), maximum marginal likelihood estimator (MMLE), inpainting, high-speed video, hyperspectral imaging**

I. Introduction

Compressive sensing (CS) [1, 2, 3, 4, 5, 6, 7, 8], which aims to recover signals from incomplete linear measurements, is based on the hypothesis that the signals in question have compressible representations. Consider, for example, a finite discrete signal $\boldsymbol{x} \in \mathbb{R}^n$, to which a sensing operator $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ ($m \ll n$) is applied to obtain a low-dimensional measurement vector $\boldsymbol{y} = \boldsymbol{\Phi x} \in \mathbb{R}^m$. Let the columns of $\mathbf{F}$ constitute a basis of $\mathbb{R}^n$. Then one can write $\boldsymbol{x} = \mathbf{F}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^n$ is a coefficient vector. The coefficients $\boldsymbol{\beta}$, and hence the signal $\boldsymbol{x}$, can be reconstructed from $\boldsymbol{y}$, if most components of $\boldsymbol{\beta}$ are negligible and $\boldsymbol{\Phi}\mathbf{F}$ satisfies certain conditions [4].

Two types of compressive representations have been used in CS: universal representations and customized representations. A universal representation is compressible for a wide range of signals. Many off-the-shelf bases provide universal compressing ability; for example, natural signals are typically compressible in discrete cosine or wavelet bases. Most CS algorithms developed to date are based on universal bases, and the versatility of the bases ensures compressibility regardless of the details of particular signals.

In contrast, a customized representation focuses on a particular collection of signals and takes into account the specifics of these signals. Customized representations can usually provide greater compressibility and allow a signal to be reconstructed from fewer measurements. It is shown in [9, 10] that significantly less measurements are required for recovering the signals from a Gaussian mixture model (GMM) by using the GMM to represent the signals than by using a universal representation, assuming the covariance matrix of each Gaussian has few dominant eigenvalues. Here the dominant eigenvectors provide a compressive representation closely related to the concept of structured sparsity [11, 12], and the representation is customized to the signals from the GMM.

To be specific, a signal $\boldsymbol{x} \in \mathbb{R}^n$ drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \mathbf{D})$ can be represented as $\boldsymbol{x} = \mathbf{F}\boldsymbol{\beta}$, where $\mathbf{F} \in \mathbb{R}^{n\times n}$ contains the eigenvectors of $\mathbf{D}$ and $\boldsymbol{\beta}$ is the coefficient vector. While traditional CS does not specify the relative importances of basis vectors, the n eigenvectors in $\mathbf{F}$ have their relative importances prescribed by the associated eigenvalues, and the set of dominant eigenvalues determines the support of $\boldsymbol{\beta}$. It has been shown in [11], the estimate of $\boldsymbol{\beta}$ from linear measurements $\boldsymbol{y}$ is the solution to the following weighted ℓ_2-norm regularized regression

$$\widehat{\boldsymbol{\beta}} = \overline{\boldsymbol{\beta}} + \arg\min_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{F}\boldsymbol{\beta}\|_2^2 + \sigma^2 \boldsymbol{\beta}' \mathbf{A}^{-1} \boldsymbol{\beta} \right\} \quad (1)$$

where σ^2 is the variance of the measurement noise, $\mathbf{A}$ is a diagonal matrix containing the eigenvalues of $\mathbf{D}$, and $\overline{\boldsymbol{\beta}}$ is a constant vector accounting for the contribution from $\boldsymbol{\mu}$. As seen from (1), the support of $\widehat{\boldsymbol{\beta}}$ is determined by the dominant eigenvalues. It can be shown that the resulting signal estimate $\widehat{\boldsymbol{x}} = \mathbf{F}\widehat{\boldsymbol{\beta}}$ is consistent with the mean-based reconstruction formula given below in (6).

Further, Renna *et al.* [13] have shown that, most natural images and videos can be well represented by a GMM with (dominantly) low-rank covariance matrices, and that any signal drawn from such a GMM can be perfectly reconstructed from r noise-free measurements if each covariance matrix of the GMM has at most r dominant eigenvalues. This is a much less stringent reconstruction condition than that prescribed by standard restricted-isometry-property (RIP) bounds [1, 4, 5]. This theoretical result is valid under the standard assumption on $\boldsymbol{\Phi}$, *i.e.*, the elements of $\boldsymbol{\Phi}$ are drawn i.i.d. from a zero-mean Gaussian distribution. It has also been shown in [13, 14] that

J. Yang, X. Liao, X. Yuan, P. Llull, D. Brady, G. Sapiro and L. Carin are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708-0291 USA. (e-mail: {jianbo.yang, xjliao, xin.yuan, patrick.llull, guillermo.sapiro, david.brady, lcarin}@duke.edu).

an optimally designed projection matrix $\mathbf{\Phi}$ can further improve the reconstruction performance.

The current GMM-based CS theories in [9, 10, 12, 13] are based on the assumption that the underlying GMM is exactly known. In practice, the GMM has to be estimated [15, 16] and the quality of the estimate depends heavily on the training signals. While the training signals should ideally have identical statistical properties as the signals being recovered, one can only find ones that have similar statistics in practice. In some applications, it is relatively easy to find good training signals; in others, however, finding good training signals is a great challenge. To solve this problem, [10, 11] proposed to use the partially recovered signals to re-train the GMM, expecting that the re-trained model will be a better representation of the underlying signals. Given a collection of measurement vectors $\{\boldsymbol{y}_i = \mathbf{\Phi}_i\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^N$ ($\boldsymbol{\epsilon}_i$ is white Gaussian noise with zero mean and variance σ^2), a set of K Gaussian signal models, $\mathcal{N}(\cdot\,|\boldsymbol{\mu}_k, \mathbf{D}_k)$ (with mean $\boldsymbol{\mu}_k$ and covariance matrix $\mathbf{D}_k$), $k = 1, \cdots, K$, are iteratively re-trained in [11] using an algorithm called Max-Max. The algorithm alternates between the following two steps:

MAP: For $i = 1, \cdots, N$, compute from $\boldsymbol{y}_i$ the maximum *a posteriori* (MAP) estimate, $(\widehat{z}_i, \widehat{\boldsymbol{x}}_i) = \arg\min_{(k,\boldsymbol{x}_i)} \mathcal{N}(\boldsymbol{y}_i|\mathbf{\Phi}_i\boldsymbol{x}_i, \sigma^2\mathbf{I})\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \mathbf{D}_k)$, where $\boldsymbol{\mu}_k$ and $\mathbf{D}_k$ are computed from the previous ML step.

ML: For $k = 1, \cdots, K$, use the subset of recovered signals from the MAP step, $\{\widehat{\boldsymbol{x}}_i : \widehat{z}_i = k\}_{i=1}^N$, as training signals to re-estimate the k-th Gaussian's parameters $(\boldsymbol{\mu}_k, \mathbf{D}_k)$.

The Max-Max algorithm was reexamined in [11] under the name "piecewise linear estimator (PLE)," and related to expectation maximization and block-sparse dictionary learning.

The GMM is supposedly a statistical representation of $\{\boldsymbol{x}_i\}_{i=1}^N$, the true signals being recovered. Employing estimates of the true signals to self-train the model is a clever way to solve the problem of finding good training signals. The quality of the self-trained model, however, hinges on the accuracy of the estimated signals $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^N$. The model is guaranteed to improve only if the estimated signals are optimal and the estimation errors are properly accounted for when using the estimated signals to re-train the GMM.

The Max-Max algorithm has several drawbacks: (i) its objective function is defined locally within each iteration, lacking a global objective across the iterations; (ii) the MAP estimation assumes each signal is exclusively associated with a single Gaussian component; for a non-Gaussian signal, which is necessarily associated with at least two Gaussians, the estimate is not globally optimal; (iii) the covariance matrices $\{\mathbf{D}_k\}_{k=1}^K$ re-estimated in the ML step lack the terms that account for the errors of the estimated signals $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^N$.

In this paper, we present an alternative approach by reformulating the self-training problem as one of maximizing the marginal likelihood of the GMM given only the measurement vectors $\{\boldsymbol{y}_i\}_{i=1}^N$, with the true signals $\{\boldsymbol{x}_i\}_{i=1}^N$ treated as latent random vectors and marginalized out of the likelihood.

The new approach employs the marginal likelihood as a global optimization objective and uses the conditional expectations of true signals as their estimates, *i.e.*, $\widehat{\boldsymbol{x}}_i = \mathbb{E}_{\boldsymbol{x}_i|\boldsymbol{y}_i}(\boldsymbol{x}_i)$, $i = 1, \cdots, N$. Our estimates are globally optimal in the sense of minimum mean squared error (MMSE), *i.e.*, $\widehat{\boldsymbol{x}}_i = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_i|\boldsymbol{y}_i}\|\boldsymbol{\theta} - \boldsymbol{x}_i\|_2^2$. Moreover, when re-training the GMM, we correct the covariance matrices by incorporating the information from the posterior covariance matrices $\mathbb{E}_{\boldsymbol{x}_i|\boldsymbol{y}_i}(\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i)'$, where $'$ denotes matrix transpose. All these features are automatically provided by pursuing rigorous expectation maximization (EM) of the marginal likelihood. As a byproduct, we also obtain an analytic posterior distribution for each unknown signal, as shown in (12).

When the signal dimensionality is large, manipulation of the GMM's covariance matrices is computationally expensive, and estimating them from highly-incomplete measurements is difficult. This motivates us to put a near-low-rank structure on the covariance matrices, *i.e.*, for $k = 1, \cdots, K$, we construct $\mathbf{D}_k = \mathbf{F}_k\mathbf{F}_k' + \eta\mathbf{I}$, where $\mathbf{F}_k \in \mathbb{R}^{n\times r_k}$ with $r_k \ll n$, $\mathbf{I}$ is an identity matrix, and $0 < \eta \ll 1$ ensures that the covariance matrices are close to low-rank and yet invertible. With the near-low-rank constraints, one manipulates $\mathbf{F}_k$ instead of $\mathbf{D}_k$, which is much more efficient since $\mathbf{F}_k$ has significantly less columns. Such a GMM is equivalent to a mixture of (low-dimensional) factor analyzers (MFA) [17, 18, 19]. Compressive sensing of a MFA has been studied in [9], where the model is estimated from training signals. In this paper, we iteratively self-train the model using the estimates of the signals being recovered, extending EM to an MFA.

In summary, we propose rigorous EM to learn a model of the unknown signals, using noisy linear measurements of the signals. No training signals are required. With the underlying signal model represented by a GMM with full or near-low-rank covariance matrices, the learning is accomplished by iterative self-training, using MMSE estimates of the signals to re-train the model and correcting the covariance matrices in each iteration. The algorithms maximize the marginal likelihood of a GMM conditional solely on the measurements, with the unknown signals marginalized out as latent random vectors. The two algorithms are respectively referred to as *maximum marginal likelihood estimator of a Gaussian mixture model (MMLE-GMM)* and *maximum marginal likelihood estimator of a mixture of factor analyzers (MMLE-MFA)*. At the end of the algorithms, the self-trained GMM or MFA yields a posterior distribution for each signal and the posterior mean provides a closed-form reconstruction.

The proposed algorithms are applied to image inpainting, compressive sensing of high-speed video, and compressive hyperspectral imaging, where a 2-D image, a 3-D video, or a 3-D hyperspectral image is partitioned into a collection of equal-sized local patches which are vectorized to constitute $\{\boldsymbol{x}_i\}_{i=1}^N$. The measurements of a hyperspectral imagery are acquired by the coded aperture snapshot spectral imager (CASSI) [20, 21], while those of a video is acquired by the coded aperture compressive temporal imager (CACTI) [22, 23, 24, 25, 26]. In both cases, the measurement take the form of $\{\boldsymbol{y}_i = \mathbf{\Phi}_i\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^N$, where $\mathbf{\Phi}_i$ is a *local* sensing operator applied to the i-th patch and $\boldsymbol{\epsilon}_i$ is noise. We present experimental results based on simulated measurements and real measurements acquired by actual hardware, and compare the performances of MMLE-

GMM and MMLE-MFA to those of state-of-the-art algorithms.

The remainder of the paper is organized as follows. We discuss signal recovery with a given GMM in Section II, and present the maximum marginal likelihood estimator for a GMM in Section III, and for an MFA in Section IV. Extensive experimental results are presented in Section V based on simulated measurements and real measurements acquired by actual hardware. Section VI concludes the paper.

II. Statistical Signal Recovery with a Given GMM

Let $\boldsymbol{x}$ be an arbitrary random signal drawn from a given mixture of K Gaussian distributions,

$$p(\boldsymbol{x}) = \textstyle\sum_{z=1}^{K} p(\boldsymbol{x}|z)p(z), \tag{2}$$

where $p(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_z, \mathbf{D}_z)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_z$ and covariance matrix $\mathbf{D}_z$ (which is assumed invertible), $z \in \{1, \cdots, K\}$ is a discrete random variable governed by $p(z) = \pi_z$.

Our goal is to recover $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is measurement noise and $\boldsymbol{\Phi}$ is a sensing operator. Assuming $\boldsymbol{\epsilon}$ is zero-mean white Gaussian, *i.e.*, $\boldsymbol{\epsilon} \sim \mathcal{N}(\cdot|\mathbf{0}, \boldsymbol{\Lambda})$ with $\boldsymbol{\Lambda}$ a diagonal matrix, we have $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{x}, \boldsymbol{\Lambda})$. Given z, one has

$$p(\boldsymbol{y}, \boldsymbol{x}|z) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{x}, \boldsymbol{\Lambda})\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_z, \mathbf{D}_z). \tag{3}$$

After expanding the rightmost side, collecting similar terms and rearranging the results, one obtains

$$p(\boldsymbol{x}, \boldsymbol{y}|z) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{x}\end{bmatrix} \middle| \begin{bmatrix}\boldsymbol{\Phi}\boldsymbol{\mu}_z\\ \boldsymbol{\mu}_z\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Lambda}^{-1} & -\boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi}\\ -\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1} & \boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi} + \mathbf{D}_z^{-1}\end{bmatrix}^{-1}\right).$$

Using the standard formulae for a multivariate Gaussian distribution, one can write from the above equation the conditional and marginal,

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}, z) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta), \mathbf{C}_z(\boldsymbol{y}, \Theta)),\\ p(\boldsymbol{y}|z) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z(\Theta)), \end{aligned} \tag{4}$$

where

$$\begin{aligned} \boldsymbol{\eta}_z(\boldsymbol{y}, \Theta) &= \boldsymbol{\mu}_z + \left(\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi} + \mathbf{D}_z^{-1}\right)^{-1}\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_z),\\ \mathbf{C}_z(\boldsymbol{y}, \Theta) &= \left(\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi} + \mathbf{D}_z^{-1}\right)^{-1}.\\ \mathbf{R}_z(\Theta) &= \left[\boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi}\left(\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}\boldsymbol{\Phi} + \mathbf{D}_z^{-1}\right)^{-1}\boldsymbol{\Phi}'\boldsymbol{\Lambda}^{-1}\right]^{-1}, \end{aligned}$$

and $\Theta \stackrel{Def.}{=} \{\boldsymbol{\Lambda}\} \cup \{\pi_k, \boldsymbol{\mu}_k, \mathbf{D}_k\}_{k=1}^K$ is the set of parameters of $p(\boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$. Note we have written $\boldsymbol{\eta}_z$ and $\mathbf{C}_z$ as functions of $(\boldsymbol{y}, \Theta)$, and $\mathbf{R}_z$ as a function of Θ, to indicate the respective dependencies.

It follows from (4) that

$$\begin{aligned} &p(z, \boldsymbol{x}, \boldsymbol{y}|\Theta)\\ &= p(z)p(\boldsymbol{x}|\boldsymbol{y}, z)p(\boldsymbol{y}|z)\\ &= \pi_z\mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta), \mathbf{C}_z(\boldsymbol{y}, \Theta))\mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z(\Theta)), \end{aligned} \tag{5}$$

After using the matrix inversion lemma to expand the inverses in the expressions of $(\boldsymbol{\eta}, \mathbf{C}, \mathbf{R})$ and performing algebraic simplification, we can rewrite

$$\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta) = \boldsymbol{\mu}_z + \mathbf{D}_z\boldsymbol{\Phi}'(\boldsymbol{\Lambda} + \boldsymbol{\Phi}\mathbf{D}_z\boldsymbol{\Phi}')^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}_z), \tag{6}$$

$$\mathbf{C}_z(\boldsymbol{y}, \Theta) = \mathbf{D}_z - \mathbf{D}_z\boldsymbol{\Phi}'(\boldsymbol{\Lambda} + \boldsymbol{\Phi}\mathbf{D}_z\boldsymbol{\Phi}')^{-1}\boldsymbol{\Phi}\mathbf{D}_z, \tag{7}$$

$$\mathbf{R}_z(\Theta) = \boldsymbol{\Lambda} + \boldsymbol{\Phi}\mathbf{D}_z\boldsymbol{\Phi}'. \tag{8}$$

It follows from (5) that the marginal distribution of $\boldsymbol{y}$ is

$$p(\boldsymbol{y}|\Theta) = \textstyle\sum_{z=1}^{K}\pi_z\mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z(\Theta)), \tag{9}$$

and the posterior distribution of $(z, \boldsymbol{x})$ given $\boldsymbol{y}$,

$$\begin{aligned} p(z, \boldsymbol{x}|\boldsymbol{y}, \Theta) &= p(z|\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y}, z)\\ &= \rho_z(\boldsymbol{y}, \Theta)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta), \mathbf{C}_z(\boldsymbol{y}, \Theta)) \end{aligned} \tag{10}$$

$$\text{where,} \qquad \rho_z(\boldsymbol{y}, \Theta) = \frac{\pi_z\mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z(\Theta))}{\sum_{l=1}^{K}\pi_l\mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{\mu}_l, \mathbf{R}_l(\Theta))}. \tag{11}$$

By (10), the posterior distribution of $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\boldsymbol{y}, \Theta) = \textstyle\sum_{z=1}^{K}\rho_z(\boldsymbol{y}, \Theta)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta), \mathbf{C}_z(\boldsymbol{y}, \Theta)), \tag{12}$$

which is a GMM with measurement-dependent mixing proportions, means, and covariance matrices.

A. Optimal Signal Estimation in the General Case

Let $\widehat{\boldsymbol{x}}(\boldsymbol{y}, \Theta)$ be an estimate of $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$, given that $\boldsymbol{x}$ is *a priori* governed by the GMM in (2), parameterized by Θ. Since $p(\boldsymbol{x}|\boldsymbol{y}, \Theta)$ is a complete characterization of the information that $\boldsymbol{y}$ carries about $\boldsymbol{x}$, the best one can achieve is to minimize the expected error between $\widehat{\boldsymbol{x}}(\boldsymbol{y}, \Theta)$ and any $\boldsymbol{x}$ drawn from $p(\boldsymbol{x}|\boldsymbol{y}, \Theta)$. Considering all instances of $\boldsymbol{y}$ drawn from $p(\boldsymbol{y}|\Theta)$, one can use the mean squared error (MSE) to measure the quality of the estimates as

$$\text{MSE}(\Theta) = \int p(\boldsymbol{y}|\Theta)\int \|(\boldsymbol{x} - \widehat{\boldsymbol{x}}(\boldsymbol{y}, \Theta))\|_2^2\, p(\boldsymbol{x}|\boldsymbol{y}, \Theta)d\boldsymbol{x}d\boldsymbol{y}.$$

Since $p(\boldsymbol{y}|\Theta) \geq 0$, one can minimize the MSE by minimizing the inner integral for each instance of $\boldsymbol{y}$ separately. The minimum MSE (MMSE) estimate, obtained from any given $\boldsymbol{y}$, is given by the posterior mean

$$\begin{aligned} &\widehat{\boldsymbol{x}}_{\text{MMSE}}(\boldsymbol{y}, \Theta) = \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y},\Theta}(\boldsymbol{x})\\ &= \textstyle\sum_{k=1}^{K}\rho_k(\boldsymbol{y}, \Theta)\int \boldsymbol{x}\,\mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_k(\boldsymbol{y}, \Theta), \mathbf{C}_k(\boldsymbol{y}, \Theta))\mathrm{d}\boldsymbol{x}\\ &= \textstyle\sum_{k=1}^{K}\rho_k(\boldsymbol{y}, \Theta)\,\boldsymbol{\eta}_k(\boldsymbol{y}, \Theta). \end{aligned} \tag{13}$$

where $\mathbb{E}_{\boldsymbol{x}|\boldsymbol{y},\Theta}(\boldsymbol{x}) = \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{y}, \Theta)\mathrm{d}\boldsymbol{x}$. The minimum mean squared error is found to be

$$\begin{aligned} &\text{MMSE}(\Theta)\\ &= \sum_{j=1}^{K}\int \rho_j\left[\text{tr}(\mathbf{C}_j) + \left\|\boldsymbol{\eta}_j - \sum_{k=1}^{K}\rho_k\boldsymbol{\eta}_k\right\|^2\right]p(\boldsymbol{y}|\Theta)d\boldsymbol{y},\\ &= \sum_{j=1}^{K}\pi_j\text{tr}(\mathbf{C}_j)\\ &\qquad + \sum_{j=1}^{K}\int \pi_j\left\|\boldsymbol{\eta}_j - \sum_{k=1}^{K}\rho_k\boldsymbol{\eta}_k\right\|^2\mathcal{N}(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{\mu}_j, \mathbf{R}_j)\,d\boldsymbol{y}, \end{aligned}$$

where we omit the dependencies of $\{\rho_j, \boldsymbol{\eta}_j\}$ on $(\boldsymbol{y}, \Theta)$ and the dependencies of $\{\mathbf{C}_j, \mathbf{R}_j\}$ on Θ; this simplified notation

applies to other variables in subsequent sections, when doing so causes no confusion. In the single-Gaussian case in which $K = 1$, the second term in the right-most side vanishes and one has $\mathrm{MMSE}(\Theta) = \mathrm{tr}(\mathbf{C}_1)$.

B. Optimal Signal Estimation when the Gaussians are Separable A Posteriori

For any given $\boldsymbol{y}$, when $\rho_{\widehat{z}} \approx 1$ and $\rho_k \approx 0$ for any $k \neq \widehat{z}$, the posterior $p(\boldsymbol{x}|\boldsymbol{y})$ is dominated by a single Gaussian component and thus the MMSE estimate can be approximated as

$$\widehat{\boldsymbol{x}}_{\mathrm{MMSE}}(\boldsymbol{y}, \Theta) \quad \approx \quad \boldsymbol{\eta}_{\widehat{z}}(\boldsymbol{y}, \Theta). \tag{14}$$

The formula in (14) was used by the Max-Max algorithm [10, 11] and related work [15] to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$, using *maximum a posteriori* (MAP) to find the best index $\widehat{z}$ (see Section III-B for details). As an alternative, one can also use marginal MAP to determine the best z, $\widehat{z} = \arg\max_z p(z|\boldsymbol{y}, \Theta) = \arg\max_k \int p(z, \boldsymbol{x}|\boldsymbol{y}, \Theta) d\boldsymbol{x}$, as has been pursued in [27] for the problem of blind deconvolution. While the integral is difficult to perform in [27], it is computed in closed-form in our GMM case, with the expression $\int p(z, \boldsymbol{x}|\boldsymbol{y}, \Theta) d\boldsymbol{x} = \rho_z(\boldsymbol{y}, \Theta)$ provided in (11).

This special case happens when the marginal Gaussian components in (9) are separable, *i.e.*, the Gaussians (counting their major probability mass) are situated far away to each other as compared with the sizes of the masses. This option was forced in [10, 11, 15, 27], motivated in part by simplifying computational complexity.

III. Learning the GMM from the Measurements

As discussed in Section I, the signal model in (2) is often not available due to the difficulty of finding training signals drawn from the same GMM as the sensed ones. In this section, we present a method for training a GMM of the unknown signals $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^n\}_{i=1}^N$, using their noisy linear measurements $\mathcal{Y} = \{\boldsymbol{y}_i = \boldsymbol{\Phi}_i \boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^N$. We assume the sensing matrices to be signal-dependent to account for generality (*i.e.*, $\boldsymbol{\Phi}_i$ depends on the signal index i). Differently from the conventional compressive sensing which considers only one signal at a time, we here consider a set of signals at the same time. This is necessary since we are estimating a GMM which characterizes the *collective* statistical properties of multiple signals (not a single signal). It is important to note that the GMM is trained on noisy linear measurements of the signals, rather than on the signals directly. Once the GMM is estimated, the signals are reconstructed using equation (13). Practical examples of $\{\boldsymbol{x}_i\}$, $\{\boldsymbol{\Phi}_i\}$, and $\{\boldsymbol{y}_i\}$ are provided in Section V.

The proposed method is based on maximum marginal likelihood (MML) estimation of the GMM parameters Θ based on $\mathcal{Y}$, with $\mathcal{X}$ marginalized out as latent variables. The MML estimate is found by solving the optimization problem,

$$\Theta_{\mathrm{MML}} = \max_{\Theta} \sum_{i=1}^{N} \ln \sum_{z_i=1}^{K} \int p(z_i, \boldsymbol{x}_i, \boldsymbol{y}_i|\Theta) d\boldsymbol{x}_i \tag{15}$$

where $p(z_i, \boldsymbol{x}_i, \boldsymbol{y}_i|\Theta)$ is given in (5). We solve the problem using expectation maximization [28], which produces a sequence of successively improved estimates, $\{\Theta^{(t)}\}_{t\geq 1}$, by alternating between the two steps (with an initialization $\Theta^{(0)}$):

- E-step: Find the posterior distribution $p(z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^{(t-1)})$, $\forall$ i, and obtain the expected complete log-likelihood,
$$\ell(\Theta|\Theta^{(t-1)}) = \textstyle\sum_{i=1}^N \mathbb{E}_{z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^{(t-1)}} \left\{\ln p(z_i, \boldsymbol{x}_i, \boldsymbol{y}_i|\Theta)\right\}.$$
- M-step: Find the improved estimate $\Theta^{(t)}$ by maximizing the expected complete log-likelihood, $\Theta^{(t)} = \arg\max_\Theta \ell(\Theta|\Theta^{(t-1)})$.

A. Technical Details

We obtain the expression of $\ell(\Theta|\Theta^{(t-1)})$ by using (5) and (10) to calculate the expectation $\mathbb{E}_{z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^{(t-1)}}(\cdot)$,

$$\begin{aligned}
\ell(\Theta|\Theta^{(t-1)}) &= \sum_{i=1}^{N} \mathbb{E}\left\{\ln \pi_{z_i} \mathcal{N}(\boldsymbol{y}_i|\boldsymbol{\Phi}_i \boldsymbol{x}_i, \boldsymbol{\Lambda}) \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \mathbf{D}_{z_i})\right\}, \\
&= \text{constant} + \sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} \ln \pi_k \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} (\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{\eta}_{ik}^{(t-1)})' \boldsymbol{\Lambda}^{-1} (\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{\eta}_{ik}^{(t-1)}) \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} \mathrm{tr}(\boldsymbol{\Lambda}^{-1} \boldsymbol{\Phi}_i \mathbf{C}_{ik}^{(t-1)} \boldsymbol{\Phi}_i') - \frac{N}{2} \ln\det(\boldsymbol{\Lambda}) \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} (\boldsymbol{\eta}_{ik}^{(t-1)} - \boldsymbol{\mu}_k)' \mathbf{D}_k^{-1} (\boldsymbol{\eta}_{ik}^{(t-1)} - \boldsymbol{\mu}_k) \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} \left[\mathrm{tr}(\mathbf{D}_k^{-1} \mathbf{C}_{ik}^{(t-1)}) + \ln\det(\mathbf{D}_k)\right],
\end{aligned} \tag{16}$$

where we have used the shorthands, $\rho_{ik}^{(t-1)} = \rho_k(\boldsymbol{y}_i, \Theta^{(t-1)})$, $\boldsymbol{\eta}_{ik}^{(t-1)} = \boldsymbol{\eta}_k(\boldsymbol{y}_i, \Theta^{(t-1)})$, $\mathbf{C}_{ik}^{(t-1)} = \mathbf{C}_k(\boldsymbol{y}_i, \Theta^{(t-1)})$, to simplify notation.

Setting to zero the gradients of $\ell(\Theta|\Theta^{(t-1)})$ with respect to $\boldsymbol{\Lambda}$, π_k, $\boldsymbol{\mu}_k$, $\mathbf{D}_k$, $k = 1, \cdots, K$, and taking into account the constraint that $\sum_{k=1}^K \pi_k = 1$, we obtain a set of equations, the solution to which gives the optimal updates, $k = 1, \cdots, K$,

$$\pi_k^{(t)} = \frac{\sum_{i=1}^N \rho_{ik}^{(t-1)}}{\sum_{k=1}^K \sum_{i=1}^N \rho_{ik}^{(t-1)}}, \tag{17}$$

$$\boldsymbol{\mu}_k^{(t)} = \frac{\sum_{i=1}^N \rho_{ik}^{(t-1)} \boldsymbol{\eta}_{ik}^{(t-1)}}{\sum_{i=1}^N \rho_{ik}^{(t-1)}}, \tag{18}$$

$$\mathbf{D}_k^{(t)} = \frac{\sum_{i=1}^N \rho_{ik}^{(t-1)} \left[(\boldsymbol{\eta}_{ik}^{(t-1)} - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{\eta}_{ik}^{(t-1)} - \boldsymbol{\mu}_k^{(t)})' + \mathbf{C}_{ik}^{(t-1)}\right]}{\sum_{i=1}^N \rho_{ik}^{(t-1)}}, \tag{19}$$

$$\begin{aligned}
\boldsymbol{\Lambda}^{(t)} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \rho_{ik}^{(t-1)} \mathrm{diag}\Big(&(\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{\eta}_{ik}^{(t-1)})(\boldsymbol{y}_i - \boldsymbol{\Phi}_i \boldsymbol{\eta}_{ik}^{(t-1)})' \\
&+ \boldsymbol{\Phi}_i \mathbf{C}_{ik}^{(t-1)} \boldsymbol{\Phi}_i' \Big),
\end{aligned} \tag{20}$$

where $\mathrm{diag}(\cdot)$ is a diagonal matrix consisting of the diagonal elements of the matrix argument.

Iterative computation of (17)-(19) constitutes the complete MMLE-GMM algorithm. The algorithm guarantees to monotonically increase $\sum_{i=1}^N \ln \sum_{k=1}^K \pi_k^{(t)} \mathcal{N}(\boldsymbol{y}_i|\boldsymbol{\Phi}\boldsymbol{\mu}_k^{(t)}, \mathbf{R}_k(\Theta^{(t)}))$,

the marginal log-likelihood function until convergence, by the general result of expectation maximization [28]. The update of $\mathbf{\Lambda}$ is optional, since $\mathbf{\Lambda}$ can often be obtained by calibrating the measurement noise, as is true in our experiments.

B. Comparison to the Max-Max Algorithm of [10, 11]

The Max-Max algorithm in [10, 11], which assumes uniform mixing proportions (*i.e.*, $\pi_k \equiv \frac{1}{K}$), are based on the following update equations:

1) (MAP) For $i = 1, \cdots, N$, compute

$$(\widehat{\boldsymbol{x}}_i^{(t-1)}, \widehat{z}_i^{(t-1)}) = \arg\max_{\boldsymbol{x},z} \ln p(z, \boldsymbol{x}|\boldsymbol{y}_i, \boldsymbol{\mu}_z^{(t-1)}, \mathbf{D}_z^{(t-1)}), \quad (21)$$

where $\widehat{\boldsymbol{x}}_i^{(t-1)}$ gives the estimate for $\boldsymbol{x}_i$ in iteration $t-1$.

2) (ML) The Gaussian parameters are updated as

$$\boldsymbol{\mu}_k^{(t)} = \frac{1}{\left|\mathcal{C}_k^{(t-1)}\right|} \sum_{i \in \mathcal{C}_k^{(t-1)}} \widehat{\boldsymbol{x}}_i^{(t-1)}, \quad (22)$$

$$\mathbf{D}_k^{(t)} = \frac{1}{\left|\mathcal{C}_k^{(t-1)}\right|} \sum_{i \in \mathcal{C}_k^{(t-1)}} (\widehat{\boldsymbol{x}}_i^{(t-1)} - \boldsymbol{\mu}_k^{(t)})(\widehat{\boldsymbol{x}}_i^{(t-1)} - \boldsymbol{\mu}_k^{(t)})', \quad (23)$$

where $\mathcal{C}_k^{(t-1)} = \{i : \widehat{z}_i^{(t-1)} = k\}$, $k = 1, \cdots, K$.

Recalling from (10) that

$$p(z, \boldsymbol{x}|\boldsymbol{y}, \Theta) = \rho_z(\boldsymbol{y}, \Theta) \mathcal{N}(\boldsymbol{x}|\boldsymbol{\eta}_z(\boldsymbol{y}, \Theta), \mathbf{C}_z(\boldsymbol{y}, \Theta)),$$

the MAP estimate of $\boldsymbol{x}_i$ in (21) can be expressed as

$$\widehat{\boldsymbol{x}}_i^{(t-1)} = \boldsymbol{\eta}_{\widehat{z}_i^{(t-1)}}(\boldsymbol{y}_i, \Theta^{(t-1)}).$$

As shown in (14), this estimate is optimal only when the posterior $p(\boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^{(t-1)})$ is dominated by a single Gaussian component indexed by $\widehat{z}_i^{(t-1)}$. Note that Max-Max uses MAP to locate the best Gaussian, while one can also use marginal MAP to find it, as discussed in Section II-B.

Two major differences between the Max-Max algorithm and the MMLE-GMM algorithm presented in Section III-A are described below.

1) MMLE-GMM assigns each signal $\boldsymbol{x}_i$ softly to all Gaussian components, while Max-Max hard assigns $\boldsymbol{x}_i$ to a single best-approximating Gaussian component. This difference leads to the different signal estimates and GMM updates. In particular, for signal estimation, MMLE-GMM achieves global MMSE in (13) by taking into account all Gaussian components, while Max-Max only achieves local MMSE in (21) within the best-approximating Gaussian component. When updating the GMM, MMLE-GMM updates each Gaussian component in (18-19) using all estimated signals, while Max-Max updates each component in (22-23) using only the signals hard assigned to the component.
2) Given that $\boldsymbol{x}_i$ is assigned to Gaussian component k, MMLE-GMM respects $\boldsymbol{x}_i$ as a Gaussian random vector, incorporating its covariances matrix $\mathbf{C}_{ik}$ into the GMM update in (19), while Max-Max approximates $\boldsymbol{x}_i$ as a point mass concentrated on the MAP estimate, ignoring its covariances matrix in the update performed in (23).

In summary, MMLE-GMM is an exact EM algorithm for MML estimation and it monotonically increases the marginal likelihood until the likelihood converges to a maxima, whereas Max-Max replaces the expectation in the E-step with MAP estimation and it is an approximate EM algorithm that has no guarantee for convergence or optimality. The exact EM has the same asymptotic time complexity as the approximate EM; however, the exact EM is practically a little slower since it has to compute the posterior covariance matrices in (7).

Essentially, the approximation of the Max-Max algorithm requires that $p(z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^{(t)})$ concentrates on the MAP estimate along the entire path $\Theta^{(0)}, \Theta^{(1)}, \Theta^{(2)}, \cdots$, which happens when the following two conditions hold: (i) for $i = 1, \cdots, N$, the probability mass of $p(z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^*)$ concentrates on the MAP estimate, where Θ^* is the optimal GMM; (ii) the initialization $\Theta^{(0)}$ is close to Θ^*. When condition (i) holds, the accuracy of the Max-Max will be dictated by how close the initialization is to Θ^*. Therefore, initialization is very important for the Max-Max algorithm; we will demonstrate this in Section V.

IV. Incorporating the Low-rank Constraints

As mentioned in Section I, directly manipulating the covariance matrices $\{\mathbf{D}_k\}$ in (2) is expensive when n is very large. To reduce the associated computational cost, we consider constrained forms for these matrices, $\mathbf{D}_k = \mathbf{F}_k\mathbf{F}_k' + \gamma\mathbf{I}$, where $\mathbf{F}_k \in \mathbb{R}^{n \times r_k}$ with $r_k \ll n$, $k = 1, \cdots, K$, and $0 < \gamma \ll 1$. Since γ is small, $\mathbf{D}_k$ is dominated by a low-rank matrix. Under these constraints, the GMM in (2) specializes to

$$p(\boldsymbol{x}) = \sum_{z=1}^{K} \pi_z \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_z, \mathbf{F}_z\mathbf{F}_z' + \gamma\mathbf{I}), \quad (24)$$

which can be equivalently written as a mixture of factor analyzers (MFA) [17, 18, 19],

$$p(\boldsymbol{x}) = \sum_{z=1}^{K} \pi_z \int \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_z + \mathbf{F}_z\boldsymbol{\beta}, \gamma\mathbf{I}) \mathcal{N}(\boldsymbol{\beta}|\mathbf{0}, \mathbf{I}) d\boldsymbol{\beta} \quad (25)$$

where $\mathbf{F}_z$ is known as the factor loading matrix of the z-th factor analyzer and $\boldsymbol{\beta}$ represents the factor scores of $\boldsymbol{x}$ (or coefficients of $\boldsymbol{x}$ in $\mathbf{F}_z$). Recalling from the discussion below (2) that $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\mathbf{\Phi}\boldsymbol{x}, \mathbf{\Lambda})$, we can write the joint distribution as

$$\begin{aligned}
&p(z, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}|\Theta) \\
&= \pi_z \mathcal{N}(\boldsymbol{y}|\mathbf{\Phi}\boldsymbol{x}, \mathbf{\Lambda}) \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_z + \mathbf{F}_z\boldsymbol{\beta}, \gamma\mathbf{I}) \mathcal{N}(\boldsymbol{\beta}|\mathbf{0}, \mathbf{I}), \\
&= \pi_z \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \\ \boldsymbol{\beta} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{\Phi}\boldsymbol{\mu}_z \\ \boldsymbol{\mu}_z \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{\Lambda}^{-1}, & -\mathbf{\Lambda}^{-1}\mathbf{\Phi}, & \mathbf{0} \\ -\mathbf{\Phi}'\mathbf{\Lambda}^{-1}, & \mathbf{\Phi}'\mathbf{\Lambda}^{-1}\mathbf{\Phi} + \frac{\mathbf{I}}{\gamma}, & -\frac{\mathbf{F}_z}{\gamma} \\ \mathbf{0}, & -\frac{\mathbf{F}_z'}{\gamma}, & \mathbf{I} + \frac{\mathbf{F}_z'\mathbf{F}_z}{\gamma} \end{bmatrix}^{-1} \right), \\
&= \pi_z \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\beta} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\xi}_z \end{bmatrix}, \begin{bmatrix} \mathbf{C}_z & \mathbf{\Omega}_z \\ \mathbf{\Omega}_z' & \mathbf{Q}_z \end{bmatrix} \right) \mathcal{N}(\boldsymbol{y}_i|\mathbf{\Phi}\boldsymbol{\mu}_k, \mathbf{R}_z), \quad (26)
\end{aligned}$$

where the parameters in the last line satisfy the equations

$$\begin{bmatrix} \mathbf{C}_z & \mathbf{\Omega}_z \\ \mathbf{\Omega}_z' & \mathbf{Q}_z \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}'\mathbf{\Lambda}^{-1}\mathbf{\Phi} + \frac{\mathbf{I}}{\gamma}, & -\frac{\mathbf{F}_z}{\gamma} \\ -\frac{\mathbf{F}_z'}{\gamma}, & \mathbf{I} + \frac{\mathbf{F}_z'\mathbf{F}_z}{\gamma} \end{bmatrix}^{-1}, \quad (27)$$

$$\begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\xi}_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_z \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_z & \mathbf{\Omega}_z \\ \mathbf{\Omega}_z' & \mathbf{Q}_z \end{bmatrix} \begin{bmatrix} -\mathbf{\Phi}'\mathbf{\Lambda}^{-1} \\ \mathbf{0} \end{bmatrix} (\boldsymbol{y} - \mathbf{\Phi}\boldsymbol{\mu}), \quad (28)$$

$$\mathbf{R}_z = \left(\mathbf{\Lambda}^{-1} - \begin{bmatrix} -\mathbf{\Lambda}^{-1}\mathbf{\Phi}, \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C}_z & \mathbf{\Omega}_z \\ \mathbf{\Omega}_z' & \mathbf{Q}_z \end{bmatrix} \begin{bmatrix} -\mathbf{\Phi}'\mathbf{\Lambda}^{-1} \\ \mathbf{0} \end{bmatrix} \right)^{-1}, \quad (29)$$

which are solved to give the parameters,

$$\mathbf{R}_z = \mathbf{\Lambda} + \gamma\mathbf{\Phi}\mathbf{\Phi}' + \mathbf{\Phi}\mathbf{F}_z\mathbf{F}_z'\mathbf{\Phi}', \quad (30)$$

$$\mathbf{C}_z = (\gamma\mathbf{I} + \mathbf{F}_z\mathbf{F}_z') - (\gamma\mathbf{I} + \mathbf{F}_z\mathbf{F}_z')\mathbf{\Phi}'\mathbf{R}_z^{-1}\mathbf{\Phi}(\gamma\mathbf{I} + \mathbf{F}_z\mathbf{F}_z'), \quad (31)$$

$$\mathbf{Q}_z = \mathbf{I} - \mathbf{F}_z'\mathbf{\Phi}'\mathbf{R}_z^{-1}\mathbf{\Phi}\mathbf{F}_z, \quad (32)$$

$$\mathbf{\Omega}_z = \mathbf{F}_z - (\gamma\mathbf{I} + \mathbf{F}_z\mathbf{F}_z')\mathbf{\Phi}'\mathbf{R}_z^{-1}\mathbf{\Phi}\mathbf{F}_z. \quad (33)$$

$$\boldsymbol{\eta}_z = \boldsymbol{\mu}_z + (\gamma\mathbf{I} + \mathbf{F}_z\mathbf{F}_z')\mathbf{\Phi}'\mathbf{R}_z^{-1}(\boldsymbol{y}_i - \mathbf{\Phi}\boldsymbol{\mu}_z), \quad (34)$$

$$\boldsymbol{\xi}_z = \mathbf{F}_z'\mathbf{\Phi}'\mathbf{R}_z^{-1}(\boldsymbol{y}_i - \mathbf{\Phi}\boldsymbol{\mu}_z). \quad (35)$$

Note that $(\mathbf{R}_z, \mathbf{C}_z, \mathbf{Q}_z, \mathbf{\Omega}_z)$ are functions of Θ and $(\boldsymbol{\eta}_z, \boldsymbol{\xi}_z)$ are functions of $(\boldsymbol{y}, \Theta)$. These dependencies are dropped in the above expressions to conserve space.

It follows from (26) that the marginal distribution of $\boldsymbol{y}$ is

$$p(\boldsymbol{y}|\Theta) = \sum_{z=1}^{K} \pi_z \mathcal{N}(\boldsymbol{y}|\mathbf{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z), \quad (36)$$

and the posterior distribution of $(z, \boldsymbol{\beta}, \boldsymbol{x})$ given $\boldsymbol{y}$,

$$p(z, \boldsymbol{\beta}, \boldsymbol{x}|\boldsymbol{y}, \Theta) = \rho_z \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\beta} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\xi}_z \end{bmatrix}, \begin{bmatrix} \mathbf{C}_z & \mathbf{\Omega}_z \\ \mathbf{\Omega}_z & \mathbf{Q}_z \end{bmatrix} \right), \quad (37)$$

$$\text{where,} \qquad \rho_z = \frac{\pi_z \mathcal{N}(\boldsymbol{y}; \mathbf{\Phi}\boldsymbol{\mu}_z, \mathbf{R}_z)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{y}; \mathbf{\Phi}\boldsymbol{\mu}_l, \mathbf{R}_l)}. \quad (38)$$

A. Learning the Constrained GMM from Measurements

Assuming the same setting as in Section III, we learn the parameters $\Theta = \{\mathbf{\Lambda}\} \cup \{\pi_k, \boldsymbol{\mu}_k, \mathbf{F}_k\}_{k=1}^K$ from $\mathcal{Y}$, by maximizing the marginal log-likelihood, $i.e.$,

$$\Theta_{\text{MML}} = \max_{\Theta} \sum_{i=1}^{N} \ln \sum_{z_i=1}^{K} \iint p(z_i, \boldsymbol{\beta}_i, \boldsymbol{x}_i, \boldsymbol{y}_i|\Theta) d\boldsymbol{x}_i d\boldsymbol{\beta}_i, \quad (39)$$

where $p(z_i, \boldsymbol{x}_i, \boldsymbol{y}_i|\Theta)$ is as given in (26). The optimization is solved by expectation maximization (EM), based on the following update equations ($t \geq 1$, starting from initial $\Theta^{(0)}$):

$$\pi_k^{(t-1)} = \frac{\sum_{i=1}^{N} \rho_{ik}^{(t-1)}}{\sum_{i=1}^{N} \sum_{k=1}^{K} \rho_{ik}^{(t-1)}}, \quad (40)$$

$$\left[\boldsymbol{\mu}_k^{(t-1)}, \mathbf{F}_k^{(t-1)} \right] = \left(\sum_{i=1}^{N} \rho_{ik}^{(t-1)} \left[\boldsymbol{\eta}_{ik}^{(t-1)}, \mathbf{\Omega}_{ik}^{(t-1)} \right] \right) \times \left(\sum_{i=1}^{N} \rho_{ik}^{(t-1)} \begin{bmatrix} 1 & (\boldsymbol{\xi}_{ik}^{(t-1)})' \\ \boldsymbol{\xi}_{ik}^{(t-1)} & \boldsymbol{\xi}_{ik}^{(t-1)}(\boldsymbol{\xi}_{ik}^{(t-1)})' + \mathbf{Q}_{ik}^{(t-1)} \end{bmatrix} \right)^{-1}, \quad (41)$$

$$\mathbf{\Lambda}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \rho_{ik}^{(t-1)} \text{diag}\Big((\boldsymbol{y}_i - \mathbf{\Phi}_i\boldsymbol{\eta}_{ik}^{(t-1)})(\boldsymbol{y}_i - \mathbf{\Phi}_i\boldsymbol{\eta}_{ik}^{(t-1)})' + \mathbf{\Phi}_i\mathbf{C}_{ik}^{(t-1)}\mathbf{\Phi}_i' \Big), \quad (42)$$

where $(\boldsymbol{\eta}_{ik}^{(t-1)}, \mathbf{\Omega}_{ik}^{(t-1)}, \boldsymbol{\xi}_{ik}^{(t-1)}, \mathbf{Q}_{ik}^{(t-1)}, \mathbf{C}_{ik}^{(t-1)})$ are given in (31)-(35) and $\rho_{ik}^{(t-1)}$ is given in (38), with the superscript $^{(t)}$ indicating the iteration number and the subscript $_i$ indexing the measurements in $\mathcal{Y}$ (recall that these variables depend on $\boldsymbol{y}_i$ and/or $\mathbf{\Phi}_i$ and thus on the index i).

Iterative computation of (40)-(42) constitutes the MMLE-MFA algorithm. The update equations can be derived using similar techniques as provided in Section III-A and the details are omitted here. The update of $\mathbf{\Lambda}$ is optional for the same reasons as stated right above Section III-B.

B. Computational Complexity

Due to the near-low-rank constraints, the posterior parameters in (30)-(35) can be computed efficiently. In particular, defining $\mathbf{\Psi}_{ik} = \mathbf{\Phi}_i\boldsymbol{F}_k$ and $\mathbf{\Delta}_i = \mathbf{\Lambda} + \gamma\mathbf{\Phi}_i\mathbf{\Phi}_i'$, we use the matrix lemma to efficiently compute the inverse $\mathbf{R}_{ik}^{-1} = (\mathbf{\Delta}_i + \mathbf{\Psi}_{ik}\mathbf{\Psi}_{ik}')^{-1} = \mathbf{\Delta}_i^{-1} - \mathbf{\Delta}_i^{-1}\mathbf{\Psi}_{ik}(\mathbf{I} + \mathbf{\Psi}_{ik}'\mathbf{\Delta}^{-1}\mathbf{\Psi}_{ik})^{-1}\mathbf{\Psi}_{ik}'\mathbf{\Delta}_i^{-1}$.

A major part of the computational cost lies in the calculation of $\mathbf{R}_{ik}^{-1}$, as shown in (31)-(35) . If we do not consider the low-rank condition, the computational complexity of the matrix inversion is $O(m^3/3 + m^2)$ using Cholesky factorization [29]. By contrast, when we consider the low-rank condition, the main cost becomes the matrix inversion $(\mathbf{I} + \mathbf{\Psi}_{ik}'\mathbf{\Delta}_i^{-1}\mathbf{\Psi}_{ik})^{-1}$, whose computational complexity is $O(r_{\max}^3/3 + r_{\max}^2)$, with $r_{\max} = \max\{r_1, \cdots, r_K\} \ll m$. In many real CS systems, including the CASSI and CACTI cameras considered in our experiments, $\mathbf{\Phi}_i\mathbf{\Phi}_i'$ is diagonal [21, 22], which makes $\mathbf{\Delta}_i = \mathbf{\Lambda} + \mathbf{\Phi}_i\mathbf{\Phi}_i'$ a diagonal matrix (recall that $\mathbf{\Lambda}$ is diagonal by definition). Moreover, $\mathbf{\Delta}_i$ is independent of the Gaussian component index k. Therefore the computational cost $\mathbf{\Delta}_i^{-1}$ is negligible.

In our experiments presented in Section V, the results on CPU time comparison validate the computational efficiency due to the low-rank constraints incorporated in the maximum marginal likelihood estimation.

V. Experimental Results

We evaluate the performance of the proposed methods, MMLE-GMM and MMLE-MFA, on image inpainting, compressive sensing (CS) of high-speed video [22], and compressive hyperspectral imaging [20, 21]. In all cases, a local sensing operator is applied at each spatial pixel location to collect certain types of information of the pixel. The information being collected is application dependent. In particular, each pixel is measured for its grayscale level in inpainting (the grayscale is either directly observed or not observed at all), for its temporal motion in high-speed videoing, and for its spectral constitution in hyperspectral imaging. The local sensing mode ensures that the Gramian of the sensing vectors for each spatial patch, say $\mathbf{\Phi}_i\mathbf{\Phi}_i'$, is diagonal, and thus the fast computation method in Section IV-B can be applied.

The evaluation is performed in comparison with other state-of-the-art methods, including KSVD-OMP [30],[1] a GMM learned from Training Patches (GMM-TP) [12], Max-Max [11],[2] Two-step Iterative Shrinkage/Thresholding (TwIST)

[1]The source code of KSVD-OMP is available at: http://www.cs.technion.ac.il/∼ronrubin/software.html

[2]The source code of Max-Max is provided by the authors of [11]

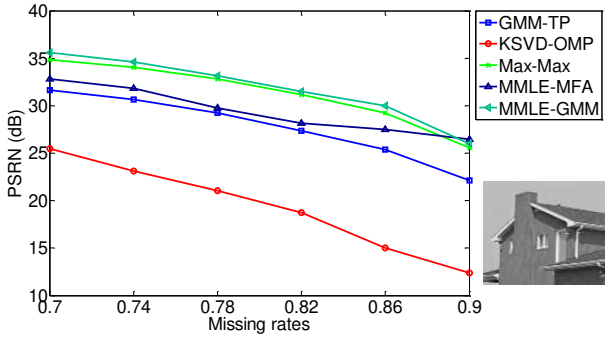

Fig. 1. The PSNRs in decibels (dB) for various methods in the image inpainting problem. The photo shows the true image.

Fig. 2. Example frames of the training video used to learn the GMM for GMM-TP, which is used to initialize Max-Max, MMLE-GMM, MMLE-MFA. The same training video is used to learn the dictionary for KSVD-OMP.

[31],[3] and generalized alternating projection (GAP) [32].[4] The performance of each method is measured by the visual quality or the peak signal-to-noise ratio (PSNR) of the reconstructions.

For MMLE-GMM, MMLE-MFA, and GMM-TP, the covariance matrix $\mathbf{\Lambda}$ for measurement noise $\boldsymbol{\epsilon}$ is assumed to be a scaled identity matrix $\mathbf{\Lambda} = \sigma^2\mathbf{I}$, with $\sigma = 10^{-4}$ given and fixed (not updated). For MMLE-MFA, the variance parameter of $p(\boldsymbol{x}|\boldsymbol{\beta})$ is assumed to be $\beta = 10^{-8}$, to indicate that each Gaussian in $p(\boldsymbol{x})$ is well approximated by a low-rank factor model. In all experiments, we run a few iterations of MMLE-GMM to get the full covariance matrices $\{\mathbf{D}_k\}_{k=1}^K$ and determine the rank r_k for MMLE-MFA as the number of dominant eigenvalues of $\mathbf{D}_k$. Note that the ranks of $\{\mathbf{D}_k\}_{k=1}^K$ converge much faster than the matrices themselves. The number of dictionary elements in KSVD is set to the best among $\{64, 128, 256, 512\}$, *i.e.*, we try these numbers one by one and report the best result. The TwIST minimizes the total-variation (TV). The GAP minimizes the weighted $\ell_{2,1}$ norm of transform coefficients, using Daubechies 4 wavelets as the spatial transform for images and video, DCT (discrete cosine transform) as the temporal transform for video, and DCT as the spatio-spectral transform for hyperspectral imagery.

A. Image Inpainting

We consider the 256×256 image shown in Figure 1. Assuming a portion of pixels are missing (due to damage, for example), the problem is to recover the missing pixels from the observed ones. We solve this problem using patch-based methods only, for which the image is partitioned into a set of overlapping 8×8 patches by sliding a 8×8 window, one pixel at a time, horizontally and vertically. Each patch is vectorized to yield a signal $\boldsymbol{x}_i$, whose measurement is simulated as $\boldsymbol{y}_i = \mathbf{\Phi}_i\boldsymbol{x}_i$, where $\mathbf{\Phi}_i$ is a diagonal matrix with diagonal elements randomly drawn from $\{0, 1\}$, with the probability of drawing 0 defined by the rate of missingness as shown in the horizontal axis of Figure 1.

An effective technique, called "synthetic basis," was proposed in [11] to initialize Max-Max. The technique generates synthetic data to best represent the image in question, and the data were used to train an initial GMM for the patches. With this initialization, Max-Max has demonstrated excellent performances in many image processing problems including inpainting. We adopt this technique to initialize Max-Max, MMLE-GMM, and MMLE-MFA.

We follow [11] to set the number of GMM components to 19 for all GMM-based methods. Following [9], we learn the dictionary for KSVD-OMP and the GMM of $p(\boldsymbol{x})$ for GMM-TP by using a training dataset constituted by 500 natural images randomly selected from the Berkeley Segmentation Dataset (http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html).

Figure 1 shows the PSNR, as a function of the fraction of missing pixels, for various reconstruction methods. It is seen that Max-Max and MMLE-GMM give the best performance, with nearly indistinguishable difference. Recall that MMLE-GMM is an exact EM algorithm while Max-Max is an approximate EM algorithm that relies heavily on good initializations. The excellent results of Max-Max demonstrate that, for image processing problems, the "synthetic basis" method can provide a good initialization for Max-Max such that the method can converge to the optimal solution.

B. Compressive Sensing of High-Speed Video

We demonstrate the efficacy of the proposed methods in compressive sensing of high-speed video, employing the coded aperture compressive temporal imaging (CACTI) system [22] to collect measurements. Each signal, say $\boldsymbol{x}_i$, is the vectorization of T consecutive $\delta \times \delta$ spatial frames, obtained by first vectorizing each frame into a column and then stacking the resulting T columns on top of each other. The measurement vector, that CACTI collects of $\boldsymbol{x}_i$, takes the form $\boldsymbol{y}_i = \mathbf{\Phi}_i\boldsymbol{x}_i$, where $\mathbf{\Phi}_i = [\mathbf{\Phi}_{i,1}, \cdots, \mathbf{\Phi}_{i,T}]$ and $\mathbf{\Phi}_{i,t}$ is a diagonal matrix with its diagonals containing the spatial masks (see below for definition) applied to the t-th frame.

We present two experiments, the first using simulated measurements, and the second using actual hardware to collect real measurements.

1) Experiment on simulated measurements: The true video contains the scenes of a basketball game. We consider 32 frames, each of the size 256×256. Each video frame is encoded with a shifted binary mask, with the mask simulated by a random binary matrix with elements drawn from the Bernoulli distribution with parameter 0.5. Every eight coded frames are collapsed into one frame of measurements by

[3]The source code of TwIST is available at http://www.lx.it.pt/∼bioucas/TwIST/TwIST.htm

[4]The source code of GAP is provided by the authors of [32]

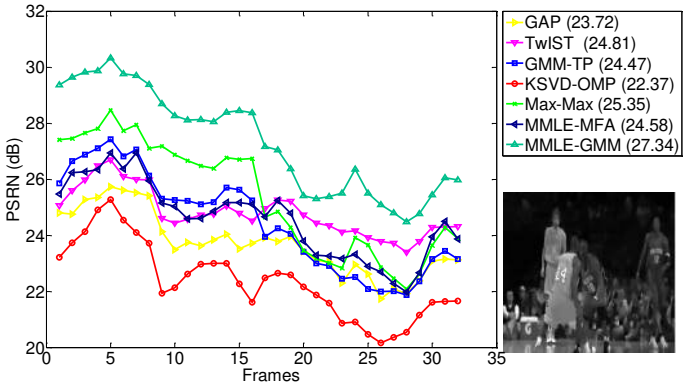

Fig. 3. Performance comparison on NBA video reconstruction. The parenthesized numbers in the legend show the PSNR averaged over all 32 frames.

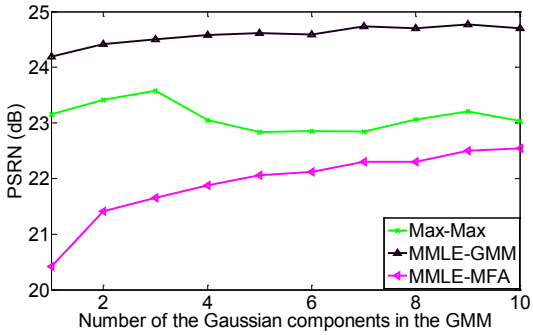

Fig. 6. The robustness of Max-Max, MMLE-GMM, and MMLE-MFA to K, demonstrated on reconstruction of a randomly selected block in the NBA video. The block contains 3721 patches.

summing each pixel's coded grayscale values over the frames, achieving a compression ratio of $1/8$. For the methods KSVD-OMP and GMM-TP, we partition each coded measurement frame into a collection of fully overlapping 4×4 patches, reconstructing a $4 \times 4 \times 8$ spatiotemporal signal from each patch.

For Max-Max, MMLE-GMM and MMLE-MFA, we partition each 256×256 measurement frame into a set of 64×64 blocks, and each block is treated as if it were a (small) regular frame and is processed independently of other blocks. Since each block is only 64×64, a small number GMM components are sufficient to capture its statistics, as shown in Figure 6. We find that the PSNR results are robust to K (the differences in PSNR are within 1 dB) as long as $2 \leq K \leq 10$. Considering the computational cost, the number of GMM components in Max-Max, MMLE-GMM, and MMLE-MFA is a random draw from $\{2, 3, 4, 5\}$ for reconstruction of each block.[5]

We use the patches of a traffic video as training data to learn a GMM for GMM-TP and use it as the initialization for MMLE-GMM, MMLE-MFA and Max-Max. The example frames of this training video are shown in Figure 2. We use the same training video to learn the dictionary for KSVD-OMP.

Figure 3 shows the PSNR's achieved by each method for the 32 video frames. The average PSNR's over the 32 frames are also shown in the brackets in the figure's legend. It is seen that MMLE-GMM and MMLE-MFA improve about 3.6dB and 1.5dB, respectively, over GMM-TP. By contrast, Max-Max performs 2dB worse than GMM-TP, although it uses exactly the same initialization. That GAP and TwIST perform worse than MMLE-GMM and MMLE-MFA demonstrates the advantages of customized dictionaries over universal bases.

2) Experiment on measurements from real hardware: We consider *real* CS measurements of fast moving letters, acquired by the CACTI system [22]. Letters are placed on the blades of a chopper wheel, that rotates at an angular velocity of 15 blades per second. The results shown here are based on 6 measurement frames that capture the fast motion of "D" during 0.15 seconds. The codes change at a rate that is 14 times as fast as the capture rate, and therefore one can reconstruct 14 video frames from a single measurement frame.

Since it has been shown in [23] that GMM-TP outperforms TwIST and GAP on this dataset, we here only compare MMLE-MFA and MMLE-GMM against GMM-TP, KSVD-OMP, and Max-Max, using the same configurations as in the experiment on simulated measurements.

Recall that training patches are in general required to learn the dictionary for KSVD-OMP and train the GMM of $p(\boldsymbol{x})$ for GMM-TP, and the trained GMM are used to initialize MMLE-MFA, MMLE-GMM and Max-Max. We investigate the influence of different training patches on the performances of the methods.

We consider two different sets of training video. The first set includes the videos of a chopper wheel rotating at several orientations, positions, and velocities. These training videos were captured by a regular camcorder at frame-rates that are different from the desired high-speed frame-rate. The second training set includes the videos of traffic scenes as illustrated in Figure 2. The first training set is deemed relevant to the video to be reconstructed, while the second is deemed irrelevant.

The high-speed frames reconstructed by the five competing methods are shown in Figure 4 for the case of relevant training video and in Figure 5 for the case of irrelevant training video. Since the true high-speed video is not available, the PSNR's cannot be computed and therefore the reconstructed video frames are evaluated by inspecting their visual quality.

As seen from Figure 4, when relevant videos are used to train $p(\boldsymbol{x})$, the frames reconstructed by GMM-TP, MMLE-GMM, and MMLE-MFA have the best quality, showing a clear chopper wheel and a sharp "D", while the frames recovered by Max-Max and KSVD-OMP are of lower quality with a significantly blurred "D". While the three best methods all have some ghosting effects, the effect seems relatively modest for MMLE-GMM.

A comparison between Figure 4 and Figure 5 shows that, when using irrelevant training videos, MMLE-GMM and MMLE-MFA can still perform satisfactorily, while GMM-TP and KSVD-OMP perform significantly worse than when using relevant training videos. These results demonstrate the advantages of the proposed marginal maximum likelihood estimators (MMLE), which learn a GMM or MFA for $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ using the noisy measurements $\mathcal{Y} = \{\boldsymbol{\Phi}_i \boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^N$. Although the

[5]More sophisticated ways of choosing K include Dirichlet processes [9], BIC/AIC [33] and the references therein. The focus of this paper is on demonstrating the basic idea of GMM-based compressive sensing. A comprehensive study of choosing K will be pursued in our future work.

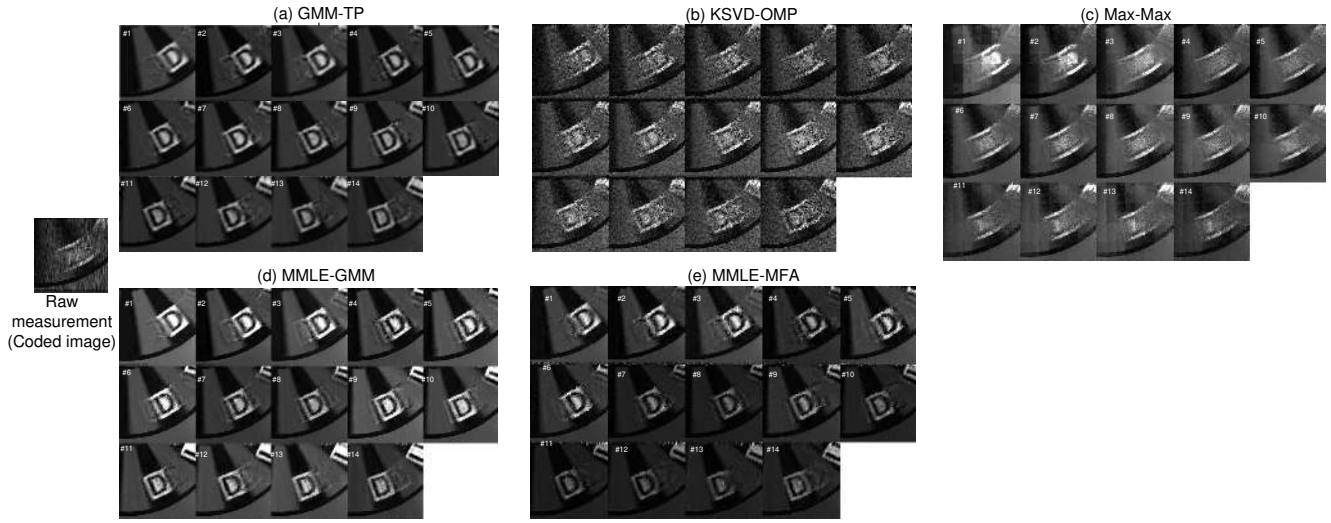

Fig. 4. Performance comparison on chopper wheel video reconstruction, when *relevant* training videos are used to learn the GMM for GMM-based methods and learn the dictionary for KSVD-OMP. Displayed from the left to the right are: the raw measurement acquired by the CACTI hardware, the frames recovered by GMM-TP (top) and MMLE-GMM (bottom), the frames recovered by KSVD-OMP (top) and MMLE-MFA (bottom), the frames recovered by Max-Max (top).

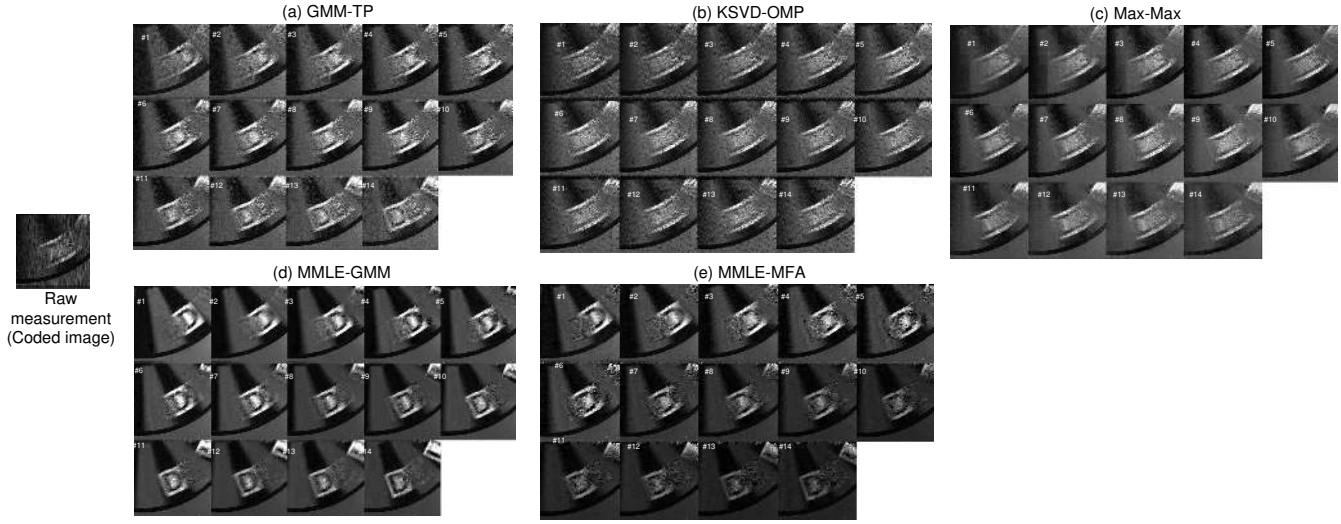

Fig. 5. Performance comparison on chopper wheel video reconstruction, when *irrelevant* training videos are used to learn the GMM for GMM-based methods and learn the dictionary for KSVD-OMP. Displayed from the left to the right are: hardware-acquired raw measurement, the frames recovered by GMM-TP (top) and MMLE-GMM (bottom), the frames recovered by KSVD-OMP (top) and MMLE-MFA (bottom), the frames recovered by Max-Max (top).

MMLE's can be initialized with a pre-trained model, this is not mandatory, since the learning is based on self-training and requires no training signals.

It is noted that Max-Max is also based on self-training. However, as discussed in Section III-B, Max-Max is an approximate EM algorithm that replaces the expectation in the E-step with MAP approximations. These approximations can make the algorithm deviate from optimality and lead to serious performance degradation. In the case when the posterior $p(z_i, \boldsymbol{x}_i|\boldsymbol{y}_i, \Theta^*)$, where Θ^* is the optimal GMM, is highly peaked at the MAP estimate for any i, and the initialization $\Theta^{(0)}$ is close to Θ^*, Max-Max could achieve excellent performance when using "synthetic basis" to obtain a good initialization, as has been observed in image inpainting in Section V-A and in other image-processing problems in [11]. Unfortunately, videos are more complicated than images because of the dynamics of the scenes they represent, which makes it more challenging to find a good initialization in the case of video reconstruction. The "synthetic basis" proposed in [11] can only be computed for images and are not applicable to videos. The relevant training videos we have used to train $\Theta^{(0)}$ are captured at frame-rates different from the desired high-speed frame-rate, and the discrepancy makes $\Theta^{(0)}$ a crude initialization unsatisfactory to Max-Max. The fact that a crude initialization is good enough for MMLE-GMM and MMLE-MFA indicates that the proposed methods are less sensitive to initialization.

TABLE I
PSNR (DB) RESULTS BY THE FOUR METHODS FOR COMPRESSIVE HYPERSPECTRAL IMAGING OF THE BIRDS.

GAP	Max-Max	MMLE-GMM	MMLE-MFA
24.16	21.48	27.26	25.53

C. Compressive Sensing of Hyperspectral Imagery

A compressive hyperspectral imager aims to reconstruct the reflectance from an object as a function of wavelength and spatial location, by measuring the coded reflectances integrated over the wavelength. In this application, each desired signal (say $\boldsymbol{x}_i$) is the vectorization of the $\delta \times \delta$ spatial frames at T consecutive wavelengths, obtained by first vectorizing each frame into a column and then stacking the resulting T columns on top of each other. The measurement vector of $\boldsymbol{x}_i$ takes the form $\boldsymbol{y}_i = \boldsymbol{\Phi}_i \boldsymbol{x}_i$, where $\boldsymbol{\Phi}_i = [\boldsymbol{\Phi}_{i,1}, \cdots, \boldsymbol{\Phi}_{i,T}]$ and $\boldsymbol{\Phi}_{i,t}$ is a diagonal matrix with its diagonals containing the spatial masks (see below for definition) applied to the frame at the t-th wavelength.

Compared with images and videos, hyperspectral imagery poses a challenge for GMM-based compressive sensing, because it is often difficult to find hyperspectral training images that have the same characteristics as those used to acquire the measurements. Thus, KSVD-OMP and GMM-TP, which require training images, often cannot be used in this problem. Assuming no training images are available, we use the minimum-norm estimates,

$$\widehat{\boldsymbol{x}}_i = \arg\min_{\boldsymbol{x}} \left\{ \|\boldsymbol{x}\|_2^2 \,:\, \boldsymbol{\Phi}_i \boldsymbol{x} = \boldsymbol{y} \right\} = \boldsymbol{\Phi}_i'(\boldsymbol{\Phi}_i \boldsymbol{\Phi}_i')^{-1} \boldsymbol{y}_i,$$

$i = 1, \cdots, N$, to train an initial GMM for MMLE-GMM, MMLE-MFA, and Max-Max.

1) Experiment on simulated measurements: We consider hyperspectral imaging of birds, whose photograph is shown in Figure 7(a). The "true" hyperspectral image is simulated as the reconstruction, obtained by TwIST, from 12 CS measurements acquired with the coded aperture snapshot spectral imaging (CASSI) camera [20, 21]. The image dataset consists of 30 spatial images, each of the size 384×512. Each spatial image corresponds to a different wavelength in the range between 450 nm and 680 nm,[6] referred to as a "spectral channel." Each spectral channel is encoded with a gray-scale mask, which is simulated by a random matrix with elements drawn from a uniform distribution in $[0, 1]$. A compressed measurement of the size 384×512 is obtained by integrating the coded voxels over the 30 spectral channels.

The experimental setup is similar to that used in video reconstruction. For MMLE-GMM, MMLE-MFA and Max-Max, we first divide the 384×512 spatial domain into a set of 64×64 blocks, and each block is divided into a collection of fully overlapping 4×4 patches. Each block of voxels are recovered independently of other blocks. For each block, the number of GMM components in Max-Max, MMLE-GMM and MMLE-MFA is a random draw from $\{2, 3, 4, 5\}$.

[6]The 30 wavelengths (nm) are: 450, 458, 465, 473, 481.5, 489.5, 498, 507, 516, 524.5, 532.5, 540.5, 548.5, 556.5, 564.5, 572.5, 580.5, 588.5, 596, 603.5, 611, 618.5, 625.8, 633.5, 641, 648.5, 656, 663.5, 671, 678.5.

Table I shows the PSNR, averaged over the 30 spectral channels, of the hyperspectral images reconstructed by each method. It is observed that MMLE-GMM and MMLE-MFA perform the best among the four methods being compared, with MMLE-GMM obtaining a 3.10dB and a 5.78db performance gain over GAP and Max-Max respectively.

Figure 7 shows the spectral patterns of the reconstructed images, with each plot computed over a dashed rectangle shown in the photo. The parenthesized numbers in the legends indicate the mean-square-errors (MSE) of the spectral curves predicted by different methods. It is observed that MMLE-MFA achieves the lowest error, followed by MMLE-GMM. These results are consistent with the PSNR results shown in Table I.

2) Experiment on hardware-acquired measurements: We consider hyperspectral imagery reconstruction from the real measurements acquired in [34] using the CASSI system [20, 21]. The system uses a single spatial light modulator (SLM) to spatially and spectrally encode the voxel volume of the object, with the coding elements taking values in $[0, 255]$. A collection of button-shaped candies (M&M's), shown in Figure 8(a), is imaged at 30 spectral channels (wavelengths),[7] which are integrated by the hardware to yield a 768×512 measurement. The measurement data are described in greater detail in [34]. The goal is to reconstruct the 30 channels from the measurement data. All methods use the same settings as in the experiment on simulated measurements.

Figure 8 plots the spectral patterns of the images reconstructed by different methods. Each curve is an average over the pixels in a dashed rectangle (associated with a particular color) shown in Figure 8(a). Following [20], we use an Ocean Optics USB2000 spectrometer to generate the reference spectrum in the figure (shown as black dots); the spectrometer measures the spectrum at selected spatial locations, for comparison with the CS-recovered results. The reference spectrum is used as an approximation of the ground truth, based on which the mean squared errors of each predicted curve is computed. The errors, shown as parenthesized numbers in the legend of Figure 8, show that MMLE-GMM gives the most accurate prediction and MMLE-MFA follows as the second best. The relative larger errors, as compared to those in Figure 7, may be attributed to the approximate "ground truth" and inaccurate calibration of the hardware. Note that the results in Figure 7 are based on simulated measurements and thus we know the exact ground truth.

D. CPU Time Comparison

We conduct numerical experiments to evaluate the CPU times of GMM-TP, KSVD-OMP, Max-Max[8], MMLE-GMM, and MMLE-MFA. The results are based on running *non-optimized* Matlab codes on the same PC, which has an Intel i5-2500 3.30GHz CPU and 16GB RAM. We run the experiments

[7]The 30 wavelengths (nm) are: 450, 458, 465, 473, 481.5, 489.5, 498, 507, 524.5, 532.5, 540.5, 548.5, 556.5, 564.5, 572.5, 580.5, 596, 603.5, 611, 618.5, 625.8, 633.5, 641, 648.5, 656, 663.5 671 and 678.5.

[8] For a fair comparison of the computational efficiency, we implemented Max-Max ourselves in this experiment to ensure that the basic matrix operations such as factorization and inversion are coded the same way for Max-Max, MMLE-GMM, and MMLE-MFA.

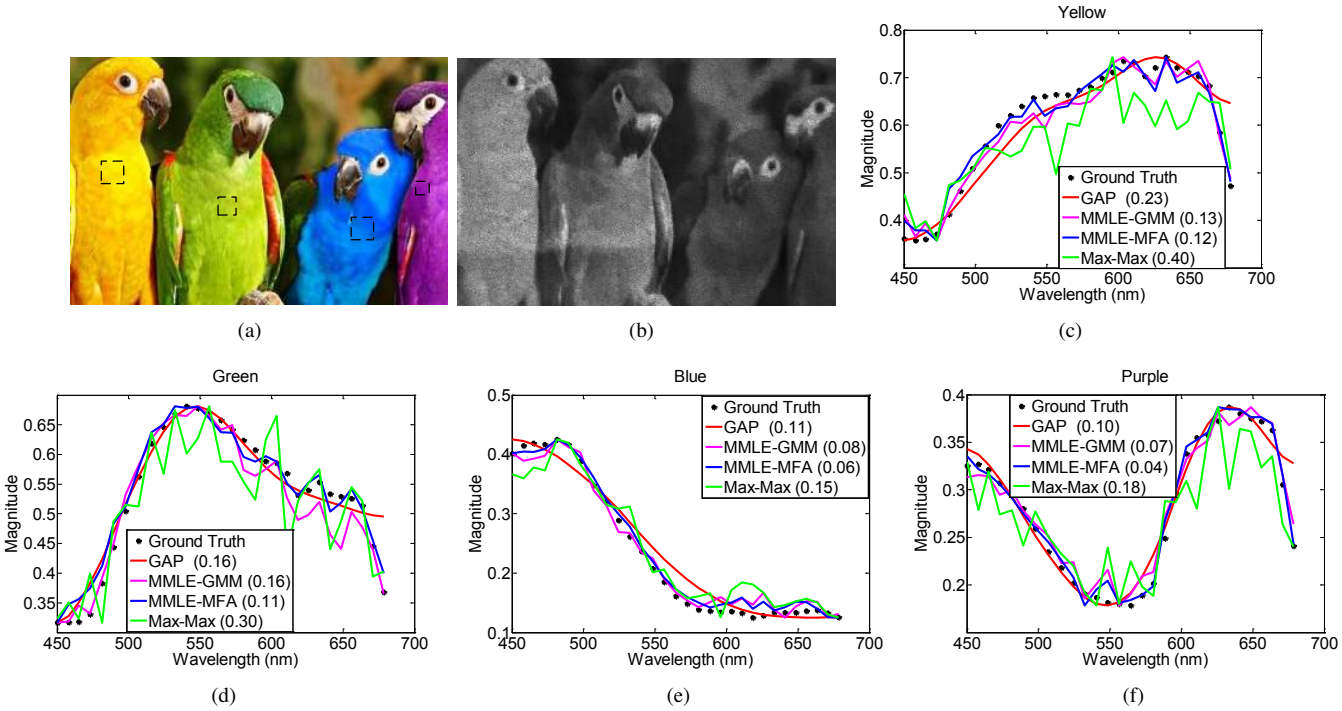

Fig. 7. Performance comparison on the birds' hyperspectral imagery reconstruction. (a) A photograph of the four birds being imaged, where a region (dashed rectangle) is chosen on each bird to represent the associated color considered in (c)-(f); (b) the simulated compressed measurement; (c)-(f) the predicted spectral patterns in the four selected regions, along with the ground truth, with the mean square error (MSE) of each method shown as a parenthesized number in the legend. The MSE's averaged over the four regions are **0.09** (MMLE-MFA), **0.12** (MMLE-GMM), **0.16** (GAP), and **0.26** (Max-Max).

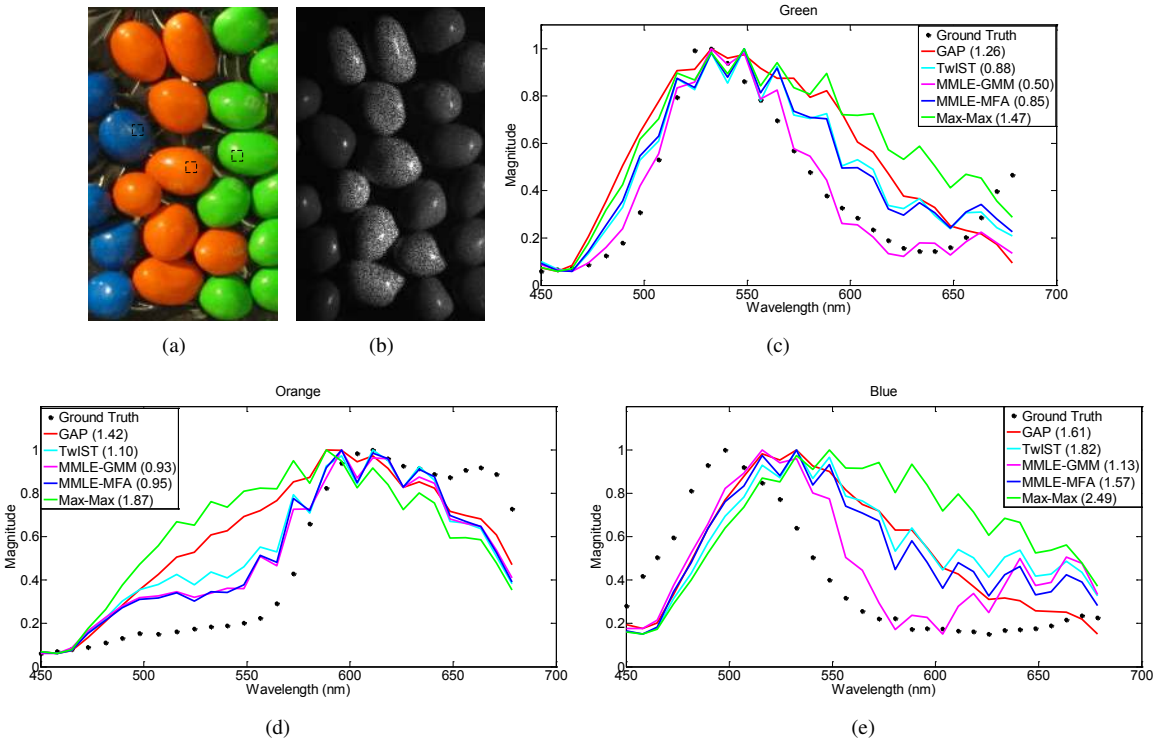

Fig. 8. Performance comparison on the M&M's hyperspectral imagery reconstruction. (a) A photograph of the M&M's being imaged, where a region (dashed rectangle) represents each color considered in (c)-(e); (b) the raw measurement acquired by the CASSI hardware; (c)-(e) the predicted spectral patterns in the three selected regions, along with the ground truth, with the mean square error (MSE) of each method shown as a parenthesized number in the legend. The MSE's averaged over the four regions are **0.85** (MMLE-GMM), **1.12** (MMLE-MFA), **1.27** (TwIST), **1.43** (GAP), and **1.94** (Max-Max).

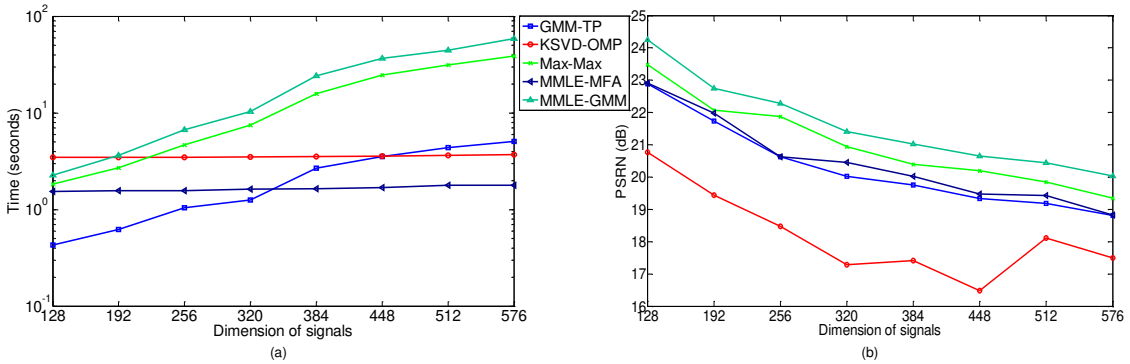

Fig. 9. (a) The CPU time and (b) the PSNR, as a function of signal dimensionality, for all methods on the reconstruction of a randomly selected block of the NBA data. The signal dimensionality shown in the horizontal axes is $4\times4\times T$, with the temporal window size T taking values in $\{8, 12, 16, 20, 24, 28, 32, 36\}$. The block contains 3721 patches. For MMLE-GMM, MMLE-MFA, and Max-Max, the time of one iteration of each respective algorithm is shown in (a). All three methods use two Gaussian components. For MMLE-MFA, the rank of each covariance matrix is set to 4. The GMM-TP uses 20 Gaussian components and reconstructs each patch by computing (12) in parallel over $z = 1, \ldots, K$. For KSVD-OMP, the number of dictionary is chosen as 128. The same training data as mentioned in Section V.B are used in GMM-TP and KSVD-OMP. For GMM-TP and KSVD-OMP, the time of signal reconstruction (excluding the time of the training procedure) is reported in (a).

five times and report the average CPU times, under various settings of the dimensions of the signals.

The results for video reconstruction are reported in Figure 9, considering the time for one block of voxels. As expected, the computational costs of all methods increase as the dimension of the signals becomes larger. However, the slopes of the timing curves for KSVD-OMP and MMLE-MFA are much smaller than those of the other three methods. It is concluded from the figure that KSVD-OMP and GMM-TP are the top two fastest methods, followed by MMLE-MFA and then by Max-Max, and MMLE-GMM is the slowest method. It is noted that MMEL-GMM, MMLE-MFA, and Max-Max are iterative methods within the EM framework. In terms of signal reconstruction accuracy, however, MMLE-GMM is the best method while KSVD-OMP and GMM-TP are the worst methods.

Among the iterative methods (within the EM framework), MMLE-MFA is the fastest method because of its explicit low-rank assumption as mentioned in Section IV-B. However, it is nontrivial to find the optimal rank for each Gaussian component in general. This issue will be further addressed in the next experiment. For Max-Max, a careful comparison of the update equations shows that Max-Max and MMLE-GMM have the same asymptotic time complexity. Nevertheless, it appears that Max-Max needs less different CPU time than MMLE-GMM in the results shown here (see Figure 9). The difference, however, reflects only the difference in coding details (MMLE need compute (5) while Max-Max need not), not the difference in theoretical time complexity.

We further evaluate the computational efficiency of MMLE-MFA, in terms of the per-iteration CPU time as a function of the maximum number of factors used in MMLE-MFA, *i.e.*, $r_{\max} = \max\{r_1, \cdots, r_K\}$. The results are shown in Figure 10. It is seen from the figure that the computational cost goes up as $r_{\max}$ increases. However, the results, in terms of the reconstruction quality evaluated by PSNR, have no clear pattern. The plausible reasons for this phenomenon include: (i) $\{r_1, \cdots, r_K\}$ are not well inferred from the data; (ii) MMLE-MFA suffers from serious local convergence.

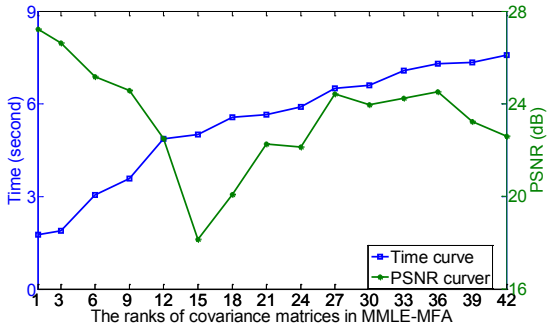

Fig. 10. The CPU time per iteration and the PSNR of MMLE-MFA, as a function of the maximum number of factors, on the reconstruction of one block of the bird hyper-spectral imagery. The block contains 3721 patches, each having 480 voxels. As a comparison, Max-Max uses 26.61 seconds and MMLE-GMM uses 36.66 seconds, and the PSNR values for these two methods are 24.97 dB and 27.40, respectively. All three methods use two Gaussian components.

VI. Conclusions

Model-based compressive sensing (CS) requires knowing the model of the target signals being measured. The signal model is usually learned from training signals, and the mismatch between the training signals and the target signals can lead to degradation of the reconstruction quality. We address this problem by learning the signal model *in situ* from the measurements of the signals in question, without resorting to other signals for training the model.

We have proposed two maximum marginal likelihood estimators (MMLE), respectively referred to as MMLE-GMM and MMLE-MFA, for learning statistical signal models from linear measurements of the signals. The MMLE-GMM, which learns a Gaussian mixture model, overcomes the two fundamental drawbacks of the Max-Max algorithm in [10, 11], by using the global MMSE signal estimates to retrain the GMM and correcting the resulting covariance matrices by adding

back an error term neglected previously. The MMLE-MFA, which learns a mixture of factor analyzers, extends MMLE-GMM to the case when the Gaussian covariance matrices have near-low-rank representations. The low-rank constraint leads to significant computational savings, outweighing the concomitant performance degradation. The marginal ML estimators use expectation maximization to achieve rigorous self-training, and enjoys guaranteed convergence and optimality. The performance gain over the Max-Max, as well as several other state-of-the-art methods, are demonstrated in an extensive set of experiments, on various problems including image inpainting, CS of high-speed video, and compressive hyperspectral imaging. The experimental results are based on both simulated data and real data acquired by actual hardware. Future work includes Bayesian approaches for regularizing ill-conditioned covariances and determining the number of Gaussian components as well as the number of factors in each component (in MMLE-MFA).

REFERENCES

[1] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions Information Theory*, vol. 52, pp. 5406–5425, 2006.

[3] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. LIX, pp. 1207–1223, 2006.

[4] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

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

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

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

[8] T. Blumensath and M. E. Davies, "Sampling theorems for signals from the union of finite-dimensional linear subspaces," *IEEE Transactions Information Theory*, vol. 55, no. 4, pp. 1872–1882, 2009.

[9] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6140–6155, December 2010.

[10] G. Yu and G. Sapiro, "Statistical compressed sensing of Gaussian mixture models," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5842–5858, December 2011.

[11] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[12] J. Yang, X. Yuan, X. Liao, P. Llull, G. Sapiro, D. J. Brady, and L. Carin, "Video compressive sensing using Gaussian mixture models," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4863–4878, 2014.

[13] F. Renna, R. Calderbank, L. Carin, and M. Rodrigues, "Reconstruction of signals drawn from a Gaussian mixture via noisy compressive measurements," *IEEE Transactions on Signal Processing*, vol. 62, no. 9, pp. 2265–2277, May 2014.

[14] W. Carson, M. Chen, M. Rodrigues, R. Calderbank, and L. Carin, "Communications inspired projection design with application to compressive sensing," *SIAM Journal on Imaging Sciences*, vol. 5, no. 4, pp. 1185–1212, 2012.

[15] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *ICCV*, 2011, pp. 479–486.

[16] ——, "Natural images, Gaussian mixtures and dead leaves," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1745–1753.

[17] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," University of Toronto, Tech. Rep., 1997.

[18] G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Transactions on Neural Networks*, vol. 8, pp. 65–74, 1997.

[19] Z. Ghahramani and M. J. Beal, "Variational Inference for Bayesian Mixtures of Factor Analysers," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 2000, pp. 449–455.

[20] D. Kittle, K. Choi, A. Wagadarikar, and D. J. Brady, "Multiframe image estimation for coded aperture snapshot spectral imagers," *Applied Optics*, vol. 49, no. 36, pp. 6824–6833, December 2010.

[21] A. Rajwade, D. Kittle, T.-H. Tsai, and L. Carin, "Coded hyperspectral imaging and blind compressive sensing," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 782–812, 2013.

[22] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. Brady, "Coded aperture compressive temporal imaging," *Optics Express*, vol. 21, no. 9, pp. 10 526–10 545, 2013.

[23] J. Yang, X. Yuan, X. Liao, P. Llull, G. Sapiro, D. J. Brady, and L. Carin, "Gaussian mixture model for video compressive sensing," *International Conference on Image Processing*, pp. 19–23, 2013.

[24] X. Yuan, J. Yang, P. Llull, X. Liao, G. Sapiro, D. J. Brady, and L. Carin, "Adaptive temporal compressive sensing for video," *International Conference on Image Processing*, pp. 14–18, 2013.

[25] X. Yuan, P. Llull, X. Liao, J. Yang, D. Brady, G. Sapiro, and L. Carin, "Low-cost compressive sensing for color video and depth," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3318–3325.

[26] P. Llull, X. Yuan, X. Liao, J. Yang, D. Brady, G. Sapiro, and L. Carin, "Compressive extended depth of field using image space coding," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2014.

[27] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2657–2664.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

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

[30] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.

[31] J. Bioucas-Dias and M. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, December 2007.

[32] X. Liao, H. Li, and L. Carin, "Generalized alternating projection for weighted-$\ell_{2,1}$ minimization with applications to model-based compressive sensing," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 797–823, 2014.

[33] R. J. Steele and A. E. Raftery, "Performance of bayesian model

selection criteria for Gaussian mixture models," University of Washington, Tech. Rep., 2008.

[34] R. Zhu, T. Tsai, and D. J. Brady, "Coded aperture snapshot spectral imager based on liquid crystal spatial light modulator," in *Frontiers in Optics*, 2013.