Dictionary Learning for Noisy and Incomplete Hyperspectral Images

[1]Zhengming Xing, [1]Mingyuan Zhou, [2]Alexey Castrodad, [2]Guillermo Sapiro and [1]Lawrence Carin

[1]Department of Electrical & Computer Engineering
Duke University, Durham, NC USA
[2]Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN, USA
POC: `lcarin@duke.edu`

Abstract

We consider analysis of noisy and incomplete hyperspectral imagery, with the objective of removing the noise and inferring the missing data. The noise statistics may be wavelength-dependent, and the fraction of data missing (at random) may be substantial, including potentially entire bands, offering the potential to significantly reduce the quantity of data that need be measured. To achieve this objective, the imagery is divided into contiguous three-dimensional (3D) spatio-spectral blocks, of spatial dimension much less than the image dimension. It is assumed that each such 3D block may be represented as a linear combination of dictionary elements of the same dimension, plus noise, and the dictionary elements are learned *in situ* based on the observed data (no *a priori* training). The number of dictionary elements needed for representation of any particular block is typically small relative to the block dimensions, and all the image blocks are processed jointly ("collaboratively") to infer the underlying dictionary. We address dictionary learning from a Bayesian perspective, considering two distinct means of imposing sparse dictionary usage. These models allow inference of the number of dictionary elements needed as well as the underlying wavelength-dependent noise statistics. It is demonstrated that drawing the dictionary elements from a Gaussian process prior, imposing structure on the wavelength dependence of the dictionary elements, yields significant advantages, relative to the more-conventional approach of using an i.i.d. Gaussian prior for the dictionary elements; this advantage is particularly evident in the presence of noise. The framework is demonstrated by processing hyperspectral imagery with a significant number of voxels missing uniformly at random, with imagery at specific wavelengths missing entirely, and in the presence of substantial additive noise.

Work partially supported by DOE, NSF, ONR, NGA, and ARO.

I. Introduction

Hyperspectral imagery (HSI) is of significant importance for many remote-sensing applications [1]–[7]. When performing such sensing, one often encounters imperfections in the data. For example, some of the voxels in the datacube may be missing, data from entire spectral bands may be missing, and the data are often contaminated with noise. The "cleaning up" of such realistic data often constitutes the first step in HSI analysis. In this paper we address these problems by utilizing new technology being developed in the field of dictionary learning for image analysis. Such dictionary-learning approaches exploit the fact that typical natural imagery may be (blockwise) expanded in terms of a linear combination of dictionary elements [8]–[19]. The low-dimensional nature of such representations makes them appropriate for addressing image imperfections of the type discussed above. Additionally, as elucidated below, one may *exploit* the ability to mitigate such imperfections to simplify hyperspectral measurements, reducing the quantity of data that need be measured in the first place ($e.g.$, *purposefully* introducing missing data within the measurement process, with the missing data recovered subsequently in the analysis).

There has been significant recent interest in sparse image representations, in the context of denoising and interpolation [8]–[15], compressive sensing (CS) [16], [20], and classification [21]. These applications exploit the fact that images may be sparsely represented in an appropriate dictionary. Recent research has demonstrated the significant utility of learning an often over-complete dictionary matched to the signals of interest ($e.g.$, images) [8]–[19], which should be contrasted with using orthonormal expansions like the discrete cosine transform or wavelets. There has been previous research on sparse representations (dictionary learning) for HSI data [22]–[26], but not in the presence of the level of missingness considered here, which as we demonstrate when comparing to other approaches, introduces significant challenges.

In addition, most of the methods for learning dictionaries are based on solving an optimization problem [8]–[14], in which one seeks to match the dictionary to the imagery of interest, while simultaneously encouraging a sparse representation. These methods have demonstrated state-of-the-art performance for denoising, super-resolution, interpolation, and inpainting. However, such methods typically assume one has access to the noise/residual variance, the size of the dictionary is set *a priori* or fixed via cross-validation, and a single ("point") estimate is learned. In HSI applications the noise variance may vary as a function of wavelength, and the wavelength-dependent noise statistics must be inferred in the analysis.

Dictionary learning has recently been cast as a factor-analysis[1] problem, with the factor loadings corresponding to the dictionary elements. The beta process (BP) [28]–[30] and the Indian buffet process (IBP) [31], [32] are nonparametric Bayesian methods well matched to estimation in factor analysis, allowing one to infer the number of factors (dictionary elements) based on the data itself. Further, one may place a prior on the noise or residual variance, with this inferred from the processed data as well [28], [29]. In this paper we extend this concept by making the noise statistics a function of the wavelength. An approximation to the full posterior density function of the model parameters may be manifested via Gibbs sampling, yielding an ensemble of dictionary representations. It has recently been demonstrated that an ensemble of solutions is often better than a single "best" solution [33] (an ensemble of multiple solutions captures uncertainty in the inference, for example based on limited data). We also compare the BP-based Bayesian construction to a generalized version of Bayesian Lasso [34], which allows further linkage of the Bayesian approach to previous optimization-based dictionary-learning methods [8]–[14].

The HSI problem has unique characteristics that should be accounted for when performing dictionary learning. One typically deals with datacubes with over 100 spectral bands, and it is expected that the image associated with most materials will be a relatively smooth function of wavelength. In the aforementioned Bayesian dictionary learning approaches, the components of the dictionary are typically drawn i.i.d. from a Gaussian distribution [29], with this corresponding to an ℓ_2 regularizer in optimization-based approaches [11], [12], as illustrated below. A contribution of this paper involves drawing the components of the dictionary from a Gaussian *process* (GP) [35], which allows one to impose a preference for dictionaries with smoothness as a function of wavelength; related smoothness constraints have been considered with non-negative matrix factorization [36]. The Bayesian formalism employed in this paper allows one to infer the GP parameters in a data-adaptive manner. The work in [37] is similar to that considered here, but it did not consider a GP, which we demonstrate significantly improves performance.

The inference of dictionary elements for representation of HSI data may be related to previous HSI research on endmembers estimation, with which HSI data have been linearly expanded [2], [38]. One distinction between the proposed model and much of the endmember research is that we model local spatial information within the dictionary, in addition to the spectral information addressed by most previous endmember research. Further, the dictionary elements are inferred in the presence of

[1] In factor analysis data are represented as a linear combination of learned basis vectors; the basis vectors are termed "factor loadings" and the weights in the superposition are called "factor scores" [27]

significant corruption to the datacube, including missing and noisy data (there might not be any "natural" endmembers in this corrupted data). Nevertheless, there are close connections between dictionary learning and endmember analysis for HSI; specifically, the idea of sparseness has been utilized widely for learning the dictionary elements [8]–[14], as well as in recent endmember research [38]–[40]. However, the explicit form of the sparseness promotion employed here is distinct from that employed previously in endmember research. For example, the beta process, when coupled with a Bernoulli process, imposes a self-consistency of the dictionary usage across the image. Additionally, within the GP we impose a prior belief about smoothness of the dictionary elements as a function of wavelength.

The remainder of the paper is organized as follows. In Section II we provide details on the problems under study, and describe the proposed dictionary learning framework for HSI data. Bayesian inference is performed with a Gibbs sampler, as discussed in Section III, and several example results are presented in Section IV based on real HSI data. Conclusions and directions for future research are discussed in Section V.

II. Bayesian Dictionary Learning Framework

Assume a hyperspectral image (HSI) is measured, and that it is partitioned into contiguous sets of voxels, with the ith set denoted $\boldsymbol{x}_i \in \mathbb{R}^{n_x \times n_y \times n_\lambda}$, where n_x and n_y represent the number of pixels in the two spatial dimensions, and n_λ represents the number of sensor wavelengths. In previous endmember research [38] one typically assumes $n_x = n_y = 1$ and consequently the signal is analyzed as a function of wavelength alone, but for our applications we have found improved performance if spatial extent is accounted for. We will also assume that there may be missing components of $\boldsymbol{x}_i$, with the missing values to be inferred in the analysis. The model is fit for voxels for which data are available, and based on the inferred model (discussed further below) the missing values are computed. For a given image we assume a set of blocks, $\{\boldsymbol{x}_i\}_{i=1,N}$, manifested by potentially considering all possible sets of (possibly overlapping) blocks. In the subsequent discussion, the $\boldsymbol{x}_i$ will be assumed represented as an "unwrapped" vector $\boldsymbol{x}_i \in \mathbb{R}^P$, with $P = n_x \cdot n_y \cdot n_\lambda$.

A. *Factor modeling for dictionary learning*

The factor model for each $\boldsymbol{x}_i$ is represented as

$$\boldsymbol{x}_i = \mathbf{D}\boldsymbol{s}_i + \boldsymbol{\epsilon}_i \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{P \times K}$ has columns that define dictionary elements, $\boldsymbol{s}_i \in \mathbb{R}^K$, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^P$ represents noise (or model residual). Note that the dictionary $\mathbf{D}$ is shared across all vector $\{\boldsymbol{x}_i\}_{i=1,N}$, and the factor score $\boldsymbol{s}_i$ is meant to be a sparse, and therefore only a subset of the dictionary elements (columns) are used to represent any particular $\boldsymbol{x}_i$. Our objective is to infer the dictionary $\mathbf{D}$ based upon all $\{\boldsymbol{x}_i\}_{i=1,N}$, and the number of employed dictionary elements (used columns of $\mathbf{D}$ for representation of $\{\boldsymbol{x}_i\}_{i=1,N}$) is anticipated to be small relative to N (and small relative to P for large n_λ). Once $\mathbf{D}$ is so learned, it may be used via the model to infer missing data, and the $\boldsymbol{\epsilon}_i$ may be subtracted out, to remove noise.

We constitute such a model in a Bayesian setting, and therefore priors are placed on the columns of $\mathbf{D}$, on the sparse vectors $\boldsymbol{s}_i$, and on the noise $\boldsymbol{\epsilon}_i$. Concerning the prior for $\boldsymbol{\epsilon}_i \in \mathbb{R}^P$, we assume the jth component of $\boldsymbol{\epsilon}_i$ may be drawn from the prior

$$\epsilon_{ij} \sim \mathcal{N}(0, \alpha^{-1}) \ , \quad \alpha \sim \text{Gamma}(a_0, b_0) \tag{2}$$

The parameters (a_0, b_0) are termed "hyper-parameters", and the gamma probability density function is represented $\text{Gamma}(\alpha; a_0, b_0) = c\alpha^{a_0-1}\text{exp}(-b_0\alpha)$, with $c = \beta^{a_0}/\Gamma(a_0)$, where is $\Gamma(\cdot)$ is a gamma function; the gamma distribution is a "conjugate" prior for α, in that given observed data drawn from the associated Gaussian distribution, the posterior of α is also gamma distributed, with updated hyperparameters [41]. The fact that such conjugate priors only require one to update hyperparameters significantly simplifies inference, as discussed further in Section III.

Note that this prior is on the *marginal* probability for each component of the noise, while the estimated posterior distribution does not assume the noise components are independent, and the full noise statistics are inferred (approximately). An important aspect of using such Bayesian constructions is that the noise statistics may be inferred (in terms of a posterior distribution on α), and need not be known *a priori*; most previous research on dictionary learning has assumed that the noise variance is known [9], [11]–[14]. Additionally, in (2) a single noise precision α is assumed associated with each $\boldsymbol{\epsilon}_i$; here we also consider the case for which a separate α_λ is assumed for the data at wavelength λ, with a separate gamma prior of the form above employed for each wavelength. This model is appropriate for the realistic case in which the noise variance is a function of wavelength.

B. Shrinkage sparseness priors and Bayesian Lasso

There are multiple ways one may impose a desire for a sparse factor score $\boldsymbol{s}_i$, and we consider two methods in this paper. The first method is based on a Bayesian form of Lasso [42]; we consider this construction with the goal of making connections with previous research on sparse dictionary learning. The Bayesian Lasso model was first developed in [34]. In this model we utilize the relationship

$$\frac{\sqrt{\gamma\alpha}}{2}\exp(-\sqrt{\gamma\alpha}|s|) = \int_0^\infty \mathcal{N}(s;0,(\alpha\xi)^{-1})\text{InvGa}(\xi;1,\gamma/2)d\xi \tag{3}$$

where InvGa$(\cdot)$ represents the inverse-gamma distribution, with InvGa$(\xi;a,b) = \frac{b^a}{\Gamma(a)}\xi^{-a-1}\exp(-b/\xi)$. Assuming for a moment that the dictionary $\mathbf{D}$ is known, we may represent a draw of the data block $\boldsymbol{x}_i$ in the following manner:

$$\begin{aligned} \boldsymbol{x}_i &\sim \mathcal{N}(\mathbf{D}\boldsymbol{s}_i, \alpha^{-1}\mathbf{I}_P) \\ s_{ik} &\sim \mathcal{N}(0, \alpha^{-1}\xi_{ik}^{-1}) \\ \alpha &\sim \text{Gamma}(a_0, b_0) \\ \xi_{ik} &\sim \text{InvGa}(1, \gamma_{ik}/2) \\ \gamma_{ik} &\sim \text{Gamma}(a_1, b_1) \end{aligned} \tag{4}$$

where $\mathbf{I}_P$ is the $P \times P$ identity matrix. Below we provide intuition for this model construction by relating it to previous optimization-based approaches.

Note that in [34] the authors considered a simpler model, in which the parameter γ_{ik} is replaced by a k-independent γ_i. The model in (4) may be viewed as a generalization of that in [34], with component-dependent hyperparameters. Similar component-dependent shrinkage has been utilized in the relevance-vector machine (RVM) [43], which employs a Student-t rather than a Laplace sparseness-promoting prior; in this case ξ_{ik} is drawn from a gamma distribution rather than an inverse-gamma, but otherwise (4) is equivalent to the RVM model. The latent variable ξ_{ik} controls whether the kth component of $\boldsymbol{s}_i$ has significant amplitude: if ξ_{ik} is large then the kth component of $\boldsymbol{s}_i$ is negligible (leading to approximately sparse vectors).

We initially consider the simplest model for the kth column of $\mathbf{D}$, $\boldsymbol{d}_k$, such that we may complete the connection of the above model to previous dictionary-learning approaches. Specifically, consider

$$\boldsymbol{d}_k \sim \mathcal{N}(0, \frac{1}{P}\mathbf{I}_P) \tag{5}$$

If we integrate out ξ_{ik} from the hierarchy in (4) using (3), the above model (applied jointly to all data

$\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1,N}$) has a log posterior density function that satisfies

$$-\log p(\alpha, \{\boldsymbol{s}_i, \boldsymbol{\gamma}_i\}_{i=1,N}|\mathcal{D}) =$$
$$\frac{\alpha}{2}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{s}_i\|_2^2 + \sum_{i=1}^{N}\sum_{k=1}^{K}\sqrt{\alpha\gamma_{ik}}|s_{ik}| + \frac{P}{2}\sum_{k=1}^{K}\|\boldsymbol{d}_k\|_2^2 + f(\alpha, \{\boldsymbol{\gamma}_i\}_{i=1,N}) \qquad (6)$$

where $f(\alpha, \{\boldsymbol{\gamma}_i\}_{i=1,N})$ is a function that captures regularization placed on α and $\{\boldsymbol{\gamma}_i\}_{i=1,N}$ by the respective gamma priors (as well as other constants). The function $f(\alpha, \{\boldsymbol{\gamma}_i\}_{i=1,N})$ essentially constrains the Lagrange multipliers (defined by α in the first term and $\sqrt{\alpha\gamma_{ik}}$ in the second) in a regularization-based solution. It is important to recognize that while the hierarchical form of the Bayesian model reflected in (4) and (5) looks somewhat unusual to those unfamiliar with such methods, the log of the posterior in the simplified model corresponds almost exactly to the form of models widely used in optimization-based inference of the model parameters [8]–[14], and it is also closely related to optimization approaches applied to learning endmembers and related HSI research [2]–[4], [38]. Specifically, the first term to the right of the equal sign in (6) corresponds to the ℓ_2 error between the data $\mathcal{D}$ and the model (which results from the Gaussian assumption on the noise and residual), the second term is a generalized ℓ_1 (Lasso) sparsifying regularizer on the dictionary weights, and the third term is a widely used smoothness term applied to the columns of the dictionary. Concerning the generalized Lasso term, with separate weights $\sqrt{\alpha\gamma_{ik}}$ on each term $|s_{ik}|$, a similar approach has been employed in *adaptive* Lasso [44].

With modern computers and numerical methods like Gibbs sampling (discussed further below), we may approximate the full posterior density function on model parameters, as opposed to a single "point" solution that maximizes (6) with respect to the dictionary weights and model parameters. In this context, we note that each consecutive density function in the hierarchical model (4) is in the conjugate-exponential family, and therefore all Gibbs update equations are analytic (see [34] for a closely related model); recall from above that conjugate priors yield updates that simply correspond to refinements of model hyperparameters, with this performed sequentially in a Gibbs sampler.

In addition to the generalized Bayesian Lasso model in (4), we also considered the original such model [34], in which instead of k-dependent γ_{ik} and ξ_{ik}, we consider k-independent γ_i and ξ_i (this also corresponds to the traditional Lasso model [42] when viewed from a MAP perspective, as in (6)). We found that this form of the Bayesian Lasso does *not* yield sparse representations in general, based upon a Gibbs sampler implementation (this is related to the inconsistency of the ℓ_1 regularization with the usual Laplace prior [45]). This can be understood by examining (4) with $\xi_{ik} \rightarrow \xi_i$, which implies that

$s_{ik} \sim \mathcal{N}(0, (\alpha\xi_i)^{-1})$ for all components k; if ξ_i is large then all s_{ik} will tend to be small, while otherwise $\boldsymbol{s}_i$ will tend not to be sparse. With the generalized Bayesian Lasso, the k-dependent ξ_{ik} allows sparseness to be manifested by favoring many of the ξ_{ik} to be large (as a function of component k), but not all of them. The generalized Bayesian Lasso is closely related to the adaptive Lasso discussed in [44], which has oracle properties.

We also considered drawing ξ_{ik} from a gamma rather than an inverse-gamma prior, thereby manifesting a fully Bayesian implementation of the RVM [43] model. We found that this RVM-like construction yields results almost identical to the generalized Bayesian Lasso model in (4); in the former the shrinkage prior is a Student-t [41], while in the latter it is the "double"-exponential in (3), and each encourages sparse dictionary weights.

C. Beta-Bernoulli sparseness priors

The generalized Bayesian Lasso construction discussed above imposes that $\{\boldsymbol{s}_i\}_{i=1,N}$ should be sparse, but it does not impose further structure (such as that the $\{\boldsymbol{s}_i\}_{i=1,N}$ should have self-consistency in which dictionary elements are used across the data $\{\boldsymbol{x}_i\}_{i=1,N}$). Further, the shrinkage prior does not impose explicit sparseness on $\boldsymbol{s}_i$, only that many of its components should be very small (but not exactly zero). Finally, the model does not allow one to directly impose a belief about the number of columns of $\mathbf{D}$ that will actually be used to represent the data ($i.e.$, although $\mathbf{D} \in \mathbb{R}^{P\times K}$, we generally set K to a large value, with the goal of automatically inferring the size of the dictionary actually used in the model). To address these goals, researchers have recently developed an Indian buffet process (IBP) [31], which may be represented in terms of the beta and Bernoulli processes [30]; this construction explicitly imposes sparseness. When presenting results, we make comparisons between the beta-Bernoulli method of this section and the generalized Bayesian Lasso model discussed in Section II-B; these are alternative means of constituting the sparse $\{\boldsymbol{s}_i\}_{i=1,N}$.

In this construction the factor scores are represented as

$$\boldsymbol{s}_i = \boldsymbol{w}_i \circ \boldsymbol{z}_i \tag{7}$$

$$\boldsymbol{w}_i \sim \mathcal{N}(0, \gamma_w^{-1}\mathbf{I}_k) \tag{8}$$

where $\boldsymbol{w}_i \in \mathbb{R}^K$, $\boldsymbol{z}_i \in \{0,1\}^K$, and $\circ$ represents the pointwise (Hadamard) vector product. The sparse binary vectors $\{\boldsymbol{z}_i\}_{i=1,N}$ are constructed via the following beta-Bernoulli process

$$z_{ik} \sim \text{Bernoulli}(\pi_k) \tag{9}$$

$$\pi_k \sim \text{Beta}(a_3/K, b_3(K-1)/K) \tag{10}$$

The Bernoulli distribution simply yields a $z_{ik} = 1$ with probability π_k, and $z_{ik} = 0$ with probability $1 - \pi_k$; the beta distribution is a prior on a continuous real random between $(0, 1)$, and is represented as $\text{Beta}(\pi; a, b) = c\pi^{a-1}(1-\pi)^{b-1}$, where $a > 0$, $b > 0$ and $c = \Gamma(a+b)/(\Gamma(a)\Gamma(b))$. In the limit $K \to \infty$ this construction reduces to a generalization of the Indian buffet process [28], [30]. In practice we truncate K, and the number of non-zero components of each $\boldsymbol{z}_i$ is a random number drawn from $\text{Binomial}(K, a_3K/(a_3 + b_3(K-1)))$, and in the limit $K \to \infty$ this reduces to $\text{Poisson}(a_3/b_3)$. We may therefore explicitly impose a prior belief on the number of dictionary elements used for each $\boldsymbol{x}_i$ (*i.e.*, the number of non-zero components in $\boldsymbol{s}_i$).

An important aspect of the above beta-Bernoulli construction is that the set of probabilities $\{\pi_k\}_{k=1,K}$ are shared for all $\{\boldsymbol{z}_i\}_{i=1,N}$, which implies that if a particular π_k is large (near one) then the associated dictionary element $\boldsymbol{d}_k$ is likely to be used to represent many of the vectors $\{\boldsymbol{x}_i\}_{i=1,N}$. Similarly, if π_k is small, then associated dictionary element is unlikely to be used across $\{\boldsymbol{x}_i\}_{i=1,N}$. Hence, the model imposes a self-consistency in the use of dictionary elements, which is well matched to the properties of many natural images [46]. This is a key property of this sparseness construction, which is not accounted for in the Bayesian Lasso model in (4).

D. Gaussian process for dictionary elements

The prior on the dictionary elements presented in (5) was considered primarily to make linkages to previous sparse dictionary-learning research, where this prior manifests a smoothness constraint from a maximum *a posteriori* (MAP) perspective. However, in the context of HSI data, we have further prior information that should be exploited. Specifically, in many cases the signature of materials is a smooth function of wavelength (at least for fine wavelength sampling, and hence large n_λ). To impose this prior knowledge more explicitly, rather than drawing the components of $\boldsymbol{d}_k$ i.i.d. from a normal distribution as in (5), we draw $\boldsymbol{d}_k$ from a Gaussian *process* (GP) [35].

For the GP construction, let $\lambda_1, \ldots, \lambda_{n_\lambda}$ represent the sensor wavelengths, in increasing order. We wish to impose that for a given spatial location, the correlation between the signal at λ_j and $\lambda_{j'}$ increases with decreasing $|\lambda_j - \lambda_{j'}|$. The GP is a natural way to do this. Specifically, for each spatial location the wavelength-dependent components of each $\boldsymbol{d}_k$ are drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}(j, j') = \boldsymbol{\Sigma}(j', j) \geq 0$ represents the correlation between the signal at wavelengths λ_j and $\lambda_{j'}$. As is customary in GP analysis, we assume the covariance matrix has the form

$$\boldsymbol{\Sigma}(j, j') = \zeta_1 \exp[-|\lambda_j - \lambda_{j'}|/\zeta_2] \tag{11}$$

Separate gamma priors may be placed on both ζ_1 and ζ_2, although in the experiments we simply set ζ_2 to promote a high probability of smoothness between consecutive wavelengths (we could also place a hyper-prior on ζ_2, but doing so one must employ Metropolis-Hastings sampling [47], as there is no analytic Gibbs update equation in this case); a gamma prior is placed on ζ_1, allowing inference of an approximate posterior distribution on this parameter. As is well known, the GP construction does not require uniform sampling of wavelength, and once inference is performed using the available data, it may be used to infer signal values at any other wavelengths (to infer the image at wavelengths for which no data are measured).

This GP-based construction is examined within the dictionary learning applied to HSI data, and it is compared to performance based upon the more-typical i.i.d. normal construction in (5). The GP prior for the dictionary will be employed both in the Bayesian Lasso sparseness construction of Section II-B and the beta-Bernoulli construction of Section II-C.

III. Computational Details

We use Gibbs sampling for the model inference. Samples from the posterior distribution of each random variable are approximated by iteratively sampling from the conditional distributions, given all the other random variables. For the beta-Bernoulli model with GP, the full likelihood is represented as

$$\begin{aligned}
&P(\mathbf{Y}, \mathbf{D}, \mathbf{W}, \mathbf{Z}, \boldsymbol{\pi}, \gamma_w, \alpha, \zeta_1) = \\
&\prod_{i=1}^{N} \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\Phi}_i \mathbf{D}(\boldsymbol{w}_i \circ \boldsymbol{z}_i), \alpha^{-1} \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i) \text{Gamma}(\alpha; a_0, b_0) \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{d}_k; 0, P^{-1}\boldsymbol{\Sigma}) \text{Gamma}(\zeta_1; a_4, b_4) \\
&\prod_{i=1}^{N} \prod_{k=1}^{K} \text{Bernoulli}(z_{ik}; \pi_k) \text{Beta}(\pi_k; a_3/K, b_3(K-1)/K) \\
&\prod_{i=1}^{N} \mathcal{N}(\boldsymbol{w}_i; \mathbf{0}, \gamma_w^{-1} \mathbf{I}_K) \text{Gamma}(\gamma_w; a_2, b_2);
\end{aligned} \tag{12}$$

where $\boldsymbol{y}_i = \boldsymbol{\Phi}_i \boldsymbol{x}_i$; if there are n_i observed voxels from $\boldsymbol{x}_i$, then $\boldsymbol{\Phi}_i \in \{0, 1\}^{n_i \times P}$, where the rows of $\boldsymbol{\Phi}_i$ are all zero except a single one, corresponding to which voxels are observed. At each iteration, the samples are drawn from the following conditional distributions.

Sampling $\boldsymbol{d}_k$:

$$p(\boldsymbol{d}_k|-) = \mathcal{N}(\boldsymbol{\mu}_{d_k}, \boldsymbol{\Sigma}_{d_k})$$

where the covariance $\boldsymbol{\Sigma}_{d_k}$and mean $\boldsymbol{\mu}_{d_k}$ can be expressed as

$$\begin{aligned} \boldsymbol{\Sigma}_{d_k} &= (P\boldsymbol{\Sigma} + \alpha \textstyle\sum_{i=1}^{N} z_{ik}^2 w_{ik}^2 \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i)^{-1} \\ \boldsymbol{\mu}_{d_k} &= \alpha \boldsymbol{\Sigma}_{d_k} \textstyle\sum_{i=1}^{N} z_{ik} w_{ik} \widetilde{\boldsymbol{x}}_i^{-k} \end{aligned}$$

where $\widetilde{\boldsymbol{x}}_i^{-k} = \boldsymbol{\Phi}_i^T \boldsymbol{y}_i - \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \mathbf{D}(\boldsymbol{w}_i \circ \boldsymbol{z}_i) + \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \boldsymbol{d}_k (w_{ik} z_{ik})$. In this and the notation below, $p(\boldsymbol{d}_k|-)$ is the probability of $\boldsymbol{d}_k$ conditioned on all other parameters being fixed to the last value in the sequence of Gibbs update equations.

Sampling z_{ik} and w_{ik}:

$$\begin{aligned} p(z_{ik}|-) &= \text{Bernoulli}(\tfrac{\widetilde{\pi_k}}{\widetilde{\pi_k} + 1 - \pi_k}) \\ p(w_{ik}|-) &= (1 - z_{ik})\mathcal{N}(0, \gamma_w^{-1}) + z_{ik} \mathcal{N}(\mu_{w_{ik}}, \Sigma_{w_{ik}}) \end{aligned}$$

where

$$\begin{aligned} \widetilde{\pi_k} &= \pi_k \exp\left(-\tfrac{\alpha}{2} w_{ik}^2 \boldsymbol{d}_k^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \boldsymbol{d}_k - 2 w_{ik} \boldsymbol{d}_k^T \widetilde{\boldsymbol{x}}_i^{-k}\right) \\ \boldsymbol{\Sigma}_{w_{ik}} &= (\gamma_w + \alpha \boldsymbol{d}_k^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \boldsymbol{d}_k)^{-1} \\ \boldsymbol{\mu}_{w_{ik}} &= \alpha \Sigma_{w_{ik}} \boldsymbol{d}_k^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \widetilde{\boldsymbol{x}}_i^{-k} \end{aligned}$$

Sampling π_k:

$$p(\pi_k|-) = \text{Beta}(a_3/K + \textstyle\sum_{i=1}^{N} z_{ik}, b_3(K-1)/K + N - \sum_{i=1}^{N} z_{ik})$$

Sampling γ_w:

$$p(\gamma_w|-) = \text{Gamma}(a_2 + KN/2, b_2 + \textstyle\sum_{i=1}^{N} \boldsymbol{w}_i^T \boldsymbol{w}_i/2)$$

Sampling α:

$$p(\alpha|-) = \text{Gamma}(a_0 + \tfrac{1}{2}\textstyle\sum_{i=1}^{N}|\mathbf{\Phi}_i||_{l_0}, b_0 + \tfrac{1}{2}\textstyle\sum_{i=1}^{N}||\mathbf{\Phi}_i^T \boldsymbol{y}_i - \mathbf{\Phi}_i^T \mathbf{\Phi}_i \mathbf{D}(\boldsymbol{w}_i \circ \boldsymbol{z}_i)||_{l_2})$$

where$||.||_{\ell_0}$ denotes the ℓ_0 norm and $\|\cdot\|_{l_2}$ denotes the ℓ_2 norm.

Sampling ζ_1:

$$p(\zeta_1|-) = \text{Gamma}(a_4 + \tfrac{PK}{2}, b_4 + \tfrac{\sum_{k=1}^{K}\boldsymbol{d}_k^T \mathbf{\Sigma}^{-1}\boldsymbol{d}_k}{2})$$

For the shrinkage-based model (Section II-B), we obtain the conditional distribution of each random variable via a similar procedure. In this case, we note that the conditional distribution of ξ_{ik} is the inverse-Gaussian distribution with parameters $\lambda' = \gamma_k$ and $\mu' = \sqrt{\frac{\gamma_k}{w_i k^2}}$. The inverse-Gaussian density function is given by

$$f(x) = \sqrt{\tfrac{\lambda'}{2\pi}} x^{-3/2} exp(-\tfrac{\lambda'(x-\mu')^2}{2(\mu')^2 x}) \ ; \ x > 0$$

The sampling method for the inverse-Gaussian distribution is discussed in [48].

IV. Examples Using Measured HSI Data

A. *Data considered and model parameter settings*

The results presented below are based on analysis of two real hyperspectral data sets, one termed Urban and the other AP Hill. The Urban scene was taken with the Hyperspectral Digital Collection Experiment (HyDICE) sensor over Copperas Cove, Texas; the data are publicly available at `http://www.agc.army.mil/hypercube/`. The APHill scene was taken with the Hyperspectral Mapper (HyMAP) over Virginia (with permission from the US Army Engineer Research and Development Center, Topographic Engineering Center, Fort Belvoir, VA). The Urban data consists of 162 spectral wavelengths and 150×150 spatial pixels, and the AP Hill data has 106 spectral bands and 300×300 spatial pixels. Both datasets have complete datacubes, and in the experiments below we perform analysis based on downsampled versions of each. Note that for each datacube we have removed water-absorption bands (this is how the data were provided to the authors for analysis).

Concerning parameter settings, the gamma priors on the precision of the noise were set as $a_0 = b_0 = 10^{-6}$, with these same hyperparameters used in all results. For the shrinkage model, the hyperparameters $a_1 = b_1 = 10^{-6}$. For the beta-Bernoulli model, we set $a_3 = 128$ and $b_3 = N/4$. In all experiments the truncation level (for the shrinkage and beta-Bernoulli model) was set at $K = 128$. For the Gaussian process, the gamma prior on ζ_1 was set to have parameters $a_4 = b_4 = 10^{-6}$. While there may appear to be a relatively large number of model parameters, these are all set in a "standard" way ($e.g.$, all gamma priors are set with the same hyperparameters, as in [43]), and there has been no tuning of any parameters. We found the beta-Bernoulli model to be particularly insensitive to the truncation level K, with almost identical results manifested for $K = 256$.

Both the shrinkage factor analysis (SFA) of Section II-B and the beta-process factor analysis (BPFA) model of Section II-C may be implemented with Gibbs sampling, with analytic update equations, as summarized briefly in Section III. Random initialization was used for all model parameters. For the results presented below we employed 100 burn-in iterations and 100 collection samples (200 total, with the first 100 discarded); while this number of samples is clearly insufficient to accurately estimate the full posterior distribution on all model parameters, it has in practice proven sufficient for estimation of mean parameters and the associated mean hyperspectral image. Specifically, the collection samples may be used to provide a mean estimate of the underlying hyperspectral data, while also providing "error bars" ($e.g.$, standard deviation). When presenting inferred images below, we present the mean inferred image. All computations were performed on a desktop computer: Intel Core$^{\text{TM}}$, 2 Duo 2.8G CPU, and 3GB RAM. For analysis of the Urban data (all 150×150 spatial pixels, and 162 wavelengths), based upon 2% of the data cube selected uniformly at random, each Gibbs iteration of the GP-based BPFA model required about 10 seconds, while each iteration of the SFA model required about 80 seconds. Note that the BPFA model has at least two advantages: (i) it is highly insensitive to the truncation level K, and (ii) it is considerably faster than the SFA model. This computational acceleration is manifested because the beta-Bernoulli construction imposes that many of the factor scores are exactly zero, and therefore when the binary indicator $z_{ik} = 0$, one need not update the associated w_{ik}. By contrast, the shrinkage prior imposes that many of the factor scores are small, but not exactly zero, and therefore without setting an (arbitrary) threshold, one must always update all of the factor scores at each Gibbs iteration. This appears to be the main advantage of the BPFA framework *vis-a-vis* SFA, as the accuracy of the results from the two models are often similar, as discussed below.

B. Recovery of missing voxels

The first experiments consider the Urban and AP Hill data, and we assume observation of 2% of the hyperspectral datacube, with observed voxels selected uniformly at random (98% of the datacube, selected uniformly at random, is either not measured or simply not used in the analysis). Results below are shown for one example such draw of observed voxels, but in the context of numerous such draws highly similar results were observed. In Figures 1 and 2 are shown recovered images at spectral bands 20 and 100 for the Urban data, and in Figures 3 and 4 the same is done for the AP Hill data. These results are based upon utilizing image blocks $\boldsymbol{x}_i$ with 4×4 spatial support. Results are shown for the beta-Bernoulli and shrinkage-based factor analysis models (BPFA and SFA, respectively), with and without the Gaussian process (GP) used as a prior for the factor loadings. When GP is not employed, the components of the factor loading are drawn i.i.d. from a normal distribution, as in (5). The missing voxels are inferred at all spectral bands simultaneously, and here we only show results at two of the spectral bands, for visualization.

While the results in Figures 1-4 appear good based upon each of the methods considered, closer inspection is required to assess modeling quality. In Figure 5 we show results for the Urban data, in which we present the results for an entire spectral signature at a representative spatial location. Results are shown with BPFA and GP-BPFA, based upon analysis with 2×2 spatial blocks, and 4×4 blocks. Note that the block size is a modeling/analysis choice, and a given block size is employed on the same (downsampled) data. Specifically, in Figure 5 we consider 5% of the datacube selected uniformly at random, and the entire downsampled datacube is analyzed jointly (although we only show spectra at one spatial location). There is a tradeoff in selecting the spatial support of the blocks. In most previous endmember research [2], [38], investigators have not considered spatial information at all. By considering 2×2 or 4×4 data blocks $\boldsymbol{x}_i$, there is an opportunity to also employ spatial information in the modeling. However, if the spatial block size becomes too large, there is a danger of increased spectral-signature contamination (mixing/blurring), as a result of containing many material types within the same block $\boldsymbol{x}_i$. As illustrated in the below results, for the data considered we have found 4×4 spatial blocks to provide a good compromise.

In Figure 5 we plot the mean inferred spectra, as well as error bars reflective of one standard deviation (estimated from the Gibbs collection samples). The GP tends to yield tighter standard deviations, and the results based upon 4×4 blocks appear to be most accurate. Note that the BPFA results (without GP) are based upon 2×2 blocks, and these results manifest high variability as a function of wavelength, particularly about spectral band 80. These qualitative observations are now made quantitative.

In Table I we summarize PSNR values computed on the entire inferred datacube, for the Urban data, with no additional additive noise (additive noise is considered below). Similar results were observed for the AP Hill data, and are omitted for brevity. In Table I we consider observing 2% and 5% of the datacube, uniformly at random. The results in this table reflect one draw of the observed data, but in considering many such draws, all inferences were consistent with this table. When observing only 2% of the datacube, there is a clear advantage to employing larger spatial blocks, the 4×4 blocks performing often significantly better than the 2×2 blocks. When observing more of the voxels, 5%, the advantage of the 4×4 blocks is present, but not as marked (note of course that the best block size depends also on the sensor spatial resolution). When observing only 2% of the datacube, there is an advantage of the GP when employed within the BPFA model. As more data are observed (5%), the necessity of the GP is less apparent in the case of no additive noise. This is expected, as the GP imposes smoothness as function of wavelength, and this prior information is of particular importance when the observed data are limited. However, when the quantity of observed data is larger, the imposition of smoothness may not be as necessary, and may even be detrimental, if the data alone are sufficient to infer appropriate factor loadings (analogous to spatial-spectral endmembers). The BPFA and SFA yield comparable results, although we have found the BPFA less sensitive to the truncation level K. In fact, it is possible that the SFA results may be improved further by tuning K, but it is anticipated that such tuning will be inappropriate in practice. Additionally, the BPFA has a significant computational advantage, as discussed above.

Note from Table I that based upon only 5% of the datacube, the 4×4 spatial blocks yield PSNR values of roughly 40 dB, which is consistent with the quality of traditional coding algorithms. The significant advantage of the method developed here is that the datacube has been reconstructed in a manner which may not require one to measure all the voxels in the first place (most traditional compression algorithms first assume access to the entire datacube).

C. Missing spectral bands

One may wish to make an inference of the spectral signature at a wavelength that was not actually measured by the sensor. This objective may be manifested if the sensor fails at a wavelength or set of

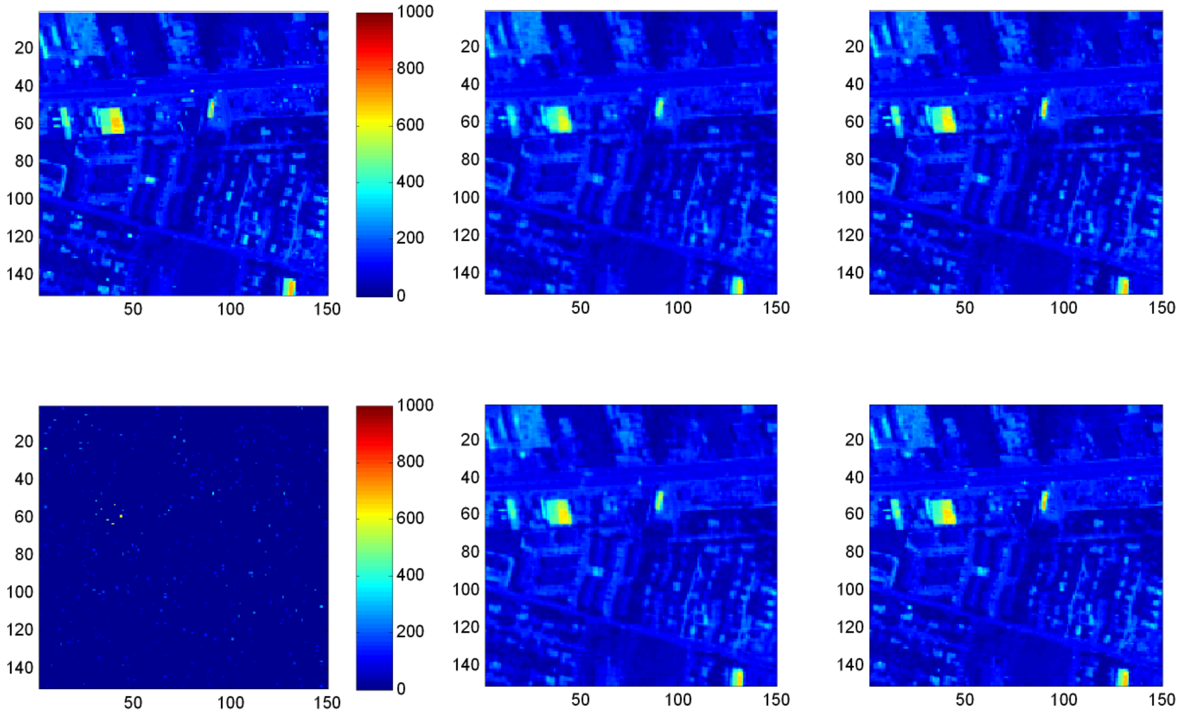

Fig. 1. Recovery of Urban hyperspectral data (normalized reflectance), based upon measuring 2% of the datacube, with voxels selected uniformly at random. The analysis is performed using 4×4 spatial blocks, and all 162 spectral bands. These results are for spectral band 20, although all spectral bands are recovered simultaneously. The same color scale is used in all images, and the total datacube is of dimension $150 \times 150 \times 162$. Results are shown for the beta-process based factor analysis (FA) model (BPFA) and for the shrinkage-based FA model (SFA), with and without a Gaussian process (GP) employed for the factor loadings. Left column: Original image for band 20 at top, and at bottom the observed data from spectral band 20 used in the analysis (unobserved pixels are here set to zero for visualization; we used similar downsampled data of this type from all spectral bands within the joint analysis). Right two columns, clockwise from top-center image: BPFA, GP-BPFA, GP-SFA, SFA.

wavelengths. This objective may also be of interest to make interpolations of the datacube, at wavelengths for which the sensor was simply not designed to sample. We would also like to make inferences about such missing spectra using a significantly downsampled hyperspectral datacube.

To examine this problem, we consider the Urban hyperspectral data, and select 16 of the 162 spectral bands at random. These 16 spectral bands are removed entirely, and a GP-BPFA analysis is performed

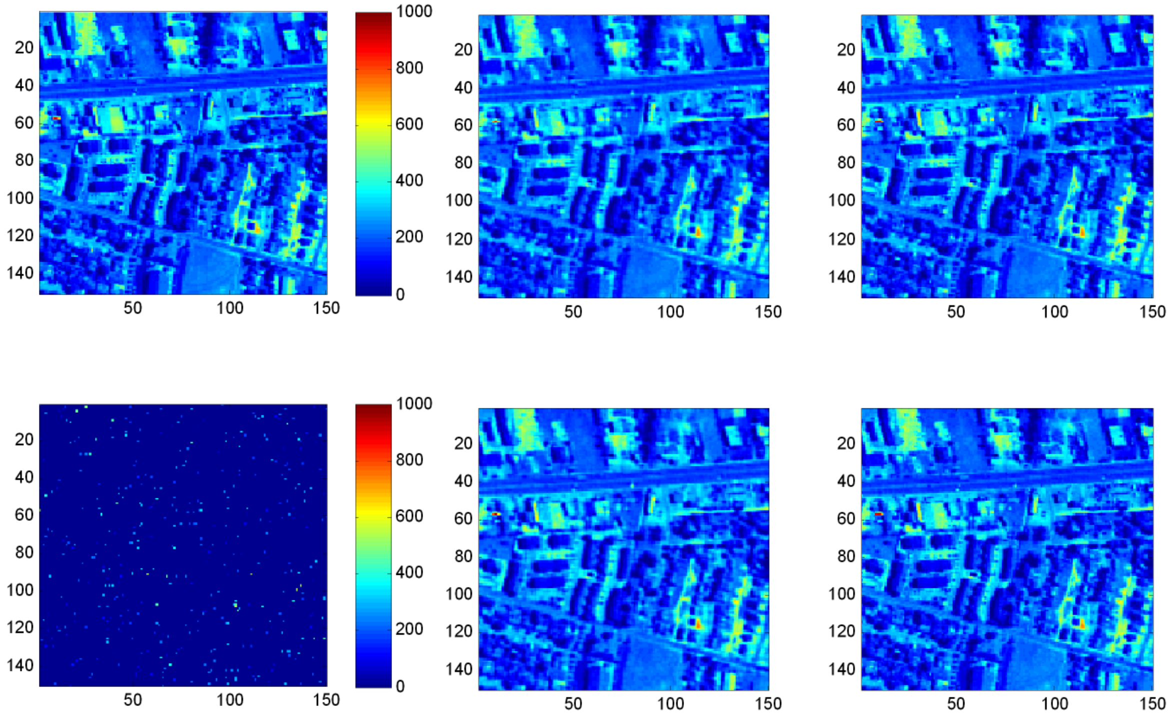

Fig. 2. Recovery of Urban hyperspectral data (normalized reflectance), based upon measuring 2% of the datacube, with voxels selected uniformly at random. The analysis is performed using 2×2 spatial blocks, and all 162 spectral bands. These results are for spectral band 100, although all spectral bands are recovered simultaneously. The subfigures are presented as in Figure 1.

using 5% of the remaining voxels, with those selected uniformly at random. The goal, essentially, is to interpolate for the missing 16 spectral bands, in the presence of massive downsampling of the datacube. Note that in this case we *must* use the GP-based formulation (the inferred GP covariance matrix, which is a continuous function of wavelength, may be used to interpolate any spectral band). Illustrative results are shown in Figure 6, in which the inferences for 2 of the 16 missing spectral bands are depicted. These results are based upon 4×4 spatial blocks, and the PSNR across all 16 missing bands is 38.3 dB.

D. Denoising

It is anticipated that in practice hyperspectral data will be noisy. In fact, the Urban and AP Hill data considered above are almost certainly undermined by sensor noise, although in the above reconstructions the evaluation of model performance was based upon the assumption that the original hyperspectral

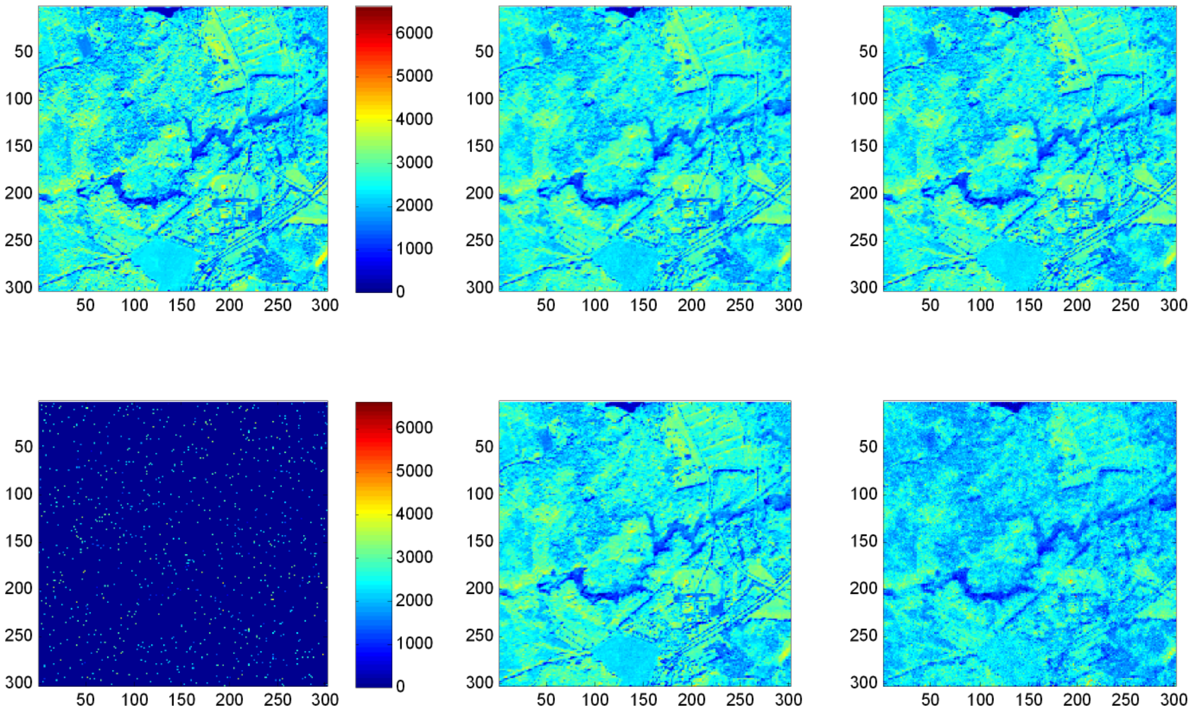

Fig. 3. Recovery of AP Hill hyperspectral data (normalized reflectance), based upon measuring 2% of the datacube, with voxels selected uniformly at random. The analysis is performed using 2×2 spatial blocks, and all 106 spectral bands. These results are for spectral band 20, although all spectral bands are recovered simultaneously. The subfigures are presented as in Figure 1.

datacubes were noise-free. We examine the noise robustness of the proposed models by now adding i.i.d. Gaussian noise to the data. We initially consider the case for which the noise variance is the same at each spectral band, but then we consider the more-realistic case for which the noise variance is wavelength dependent. Additionally, it is possible that the noise variance may vary as a function of spatial position, although that is not considered within these examples. We note, however, that although the proposed models assume an i.i.d Gaussian *prior*, with potentially wavelength-dependent variance, if spatial dependence is also manifested in the actual noise statistics, this should be approximated via the posterior density function on the noise statistics, which need not be Gaussian or stationary.

In Table II we present results for which the noise standard deviation at each spectral band is either 5,

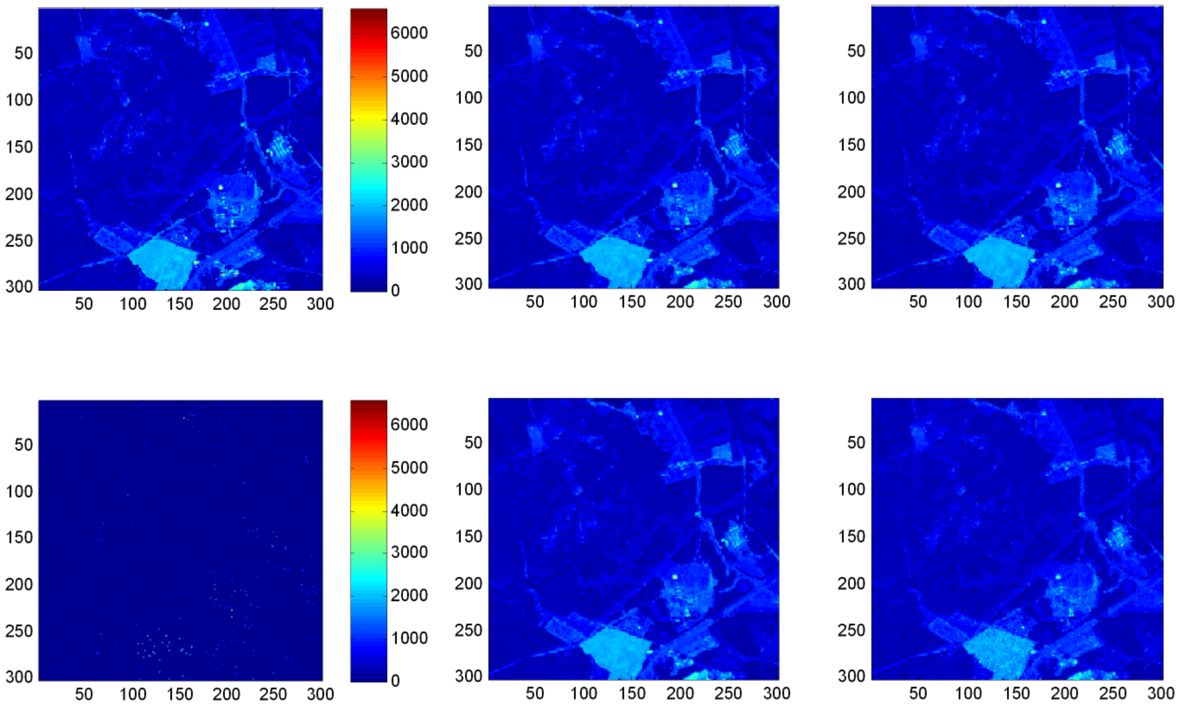

Fig. 4. Recovery of AP Hill hyperspectral data (normalized reflectance), based upon measuring 2% of the datacube, with voxels selected uniformly at random. The analysis is performed using 2×2 spatial blocks, and all 106 spectral bands. These results are for spectral band 100, although all spectral bands are recovered simultaneously. The subfigures are presented as in Figure 1.

15, 25, 35 or 50. These results are for the Urban data, with similar results manifested for the AP Hill data, omitted for brevity. All of these results employ the GP, as this was found to be essential for the noisy data. Specifically, the imposition of smoothness in the factor loadings across wavelength plays an important role in mitigating noise. In Table II we show results based upon observing 2% and 5% of the datacube, uniformly at random, and these results are based upon analyzing 4×4 spatial blocks; the larger spatial blocks, relative to 2×2, also played an important role in enhancing robustness to noise. In these results we present the PSNR value, for GP-BPFA and GP-SFA, as a function of the noise standard deviation, and we also present the mean estimate for the noise standard deviation. Note that for standard deviations in excess of 5 the models infer the underlying noise standard deviation with high accuracy.

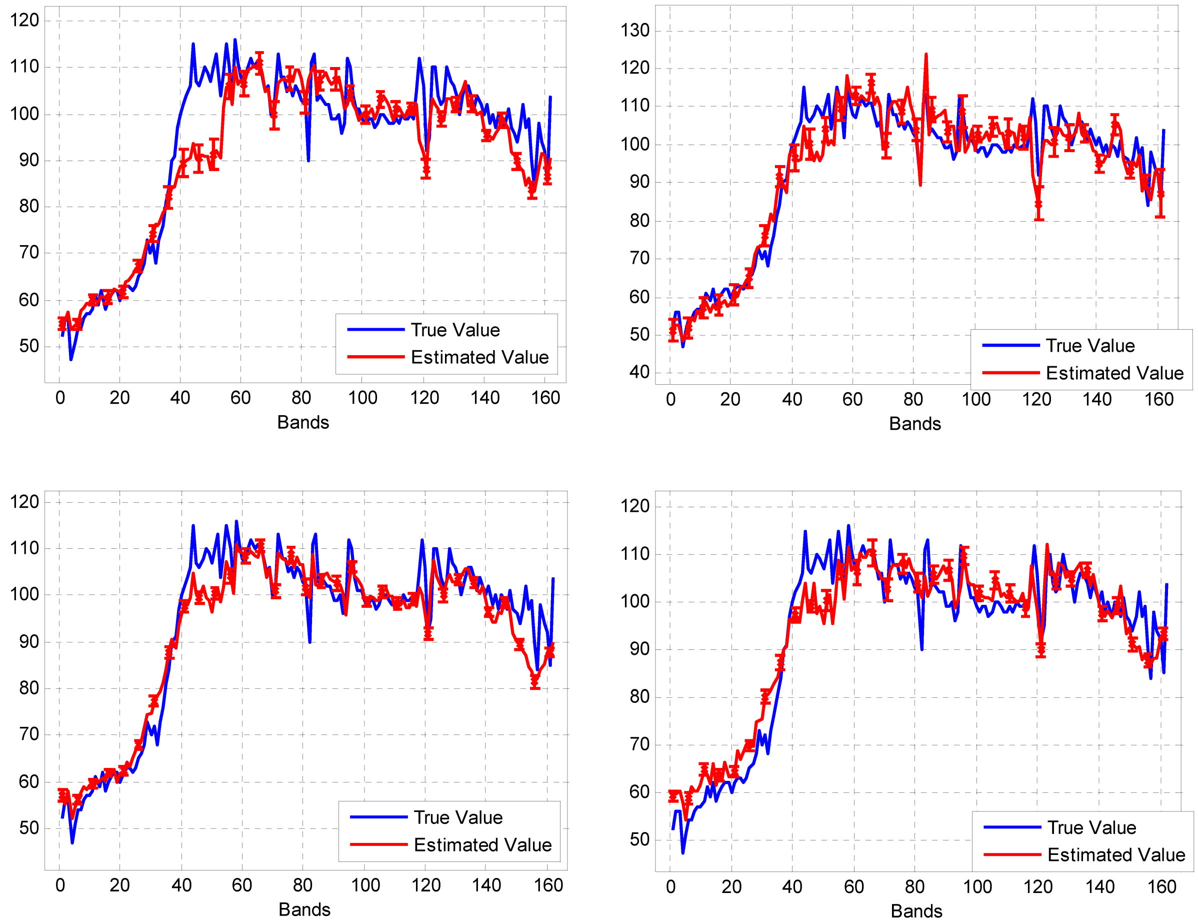

Fig. 5. Representative wavelength-dependent signature (normalized reflectance) at one spatial location, for the Urban hyperspectral data. The top row is based upon recovery using 2×2 spatial patches, and the bottom row uses 4×4 spatial patches. In all cases the same data were used for analysis, based upon selecting 5% of the voxels in the datacube uniformly at random. The left column corresponds to results based upon GP-BPFA, and the right column is BPFA.

It is more realistic to expect the noise standard deviation to vary as a function of wavelength. To examine this case we again considered the Urban data, but now the noise standard deviation for each wavelength is drawn from $\text{Gamma}(75, 1/3)$, which has 25 for a mean and a 8.3 variance. We again only consider BPFA and SFA with GP, as the GP prior on the factor loadings was found to be essential to achieving noise robustness in this case. The results in Table III consider 2×2 and 4×4 spatial blocks in the $\boldsymbol{x}_i$, and we consider cases for which 5% to 20% of the datacube is observed, uniformly at random. The results of GP-BPFA and GP-SFA are comparable, but as discussed above the former has significant

TABLE I

ACCURACY OF RECOVERED DATACUBE (PSNR, IN DB), BASED ON OBSERVING 2% AND 5% OF THE VOXELS, SELECTED UNIFORMLY AT RANDOM. RESULTS ARE SHOWN FOR THE URBAN DATA, CONSIDERING ANALYSIS WITH 2×2 AND 4×4 SPATIAL BLOCKS. RESULTS ARE SHOWN FOR THE BETA-PROCESS BASED FACTOR ANALYSIS (FA) MODEL (BPFA) AND FOR THE SHRINKAGE-BASED FA MODEL (SFA), IN EACH CASE WITH AND WITHOUT A GAUSSIAN PROCESS (GP) EMPLOYED FOR THE FACTOR LOADINGS. THERE IS NO ADDITIVE NOISE IN THIS CASE (PROCESSING ORIGINAL DATACUBE).

	2×2, 2%	4×4, 2%	2×2, 5%	4×4, 5%
BPFA	26.4	30.4	38.9	39.4
GP-BPFA	31.4	33.1	39.5	40.2
SFA	30.6	31.3	39.8	41.3
GP-SFA	29.4	33.5	38.0	41.2

advantages with regard to computational speed and robustness to setting the truncation K. The 4×4 blocks provide typically 1 dB better PSNR values relative to 2×2, and hence such block sizes are recommended.

In addition to estimating the underlying datacube, and hence denoising, the model also infers the underlying noise statistics. In Figure 7 we present the true and inferred noise standard deviation for the GP-BPFA; the results for GP-SFA are very similar, and are omitted for brevity. Results are shown for 4×4 blocks. The model infers mean wavelength-dependent noise standard deviations, and via the collection samples we also present standard deviations on the estimates. Both the GP-BPFA and GP-SFA perform this task well, and we underscore that without the GP both models failed in this task with 2% observations.

E. Related Algorithms

Recall that $\boldsymbol{x}_i \in \mathbb{R}^{n_x \times n_y \times n_\lambda}$ represents the pixels in the ith patch of given HSI data. This may be "unwrapped" to a vector $\boldsymbol{x}_i \in \mathbb{R}^P$, with $P = n_x n_y n_\lambda$. If one has access to N (overlapping) patches from given HSI data, we may let the $\boldsymbol{x}_i$ define N columns in a matrix $\mathbf{X} \in \mathbb{R}^{P \times N}$. For example, when $n_x = n_y = 2$, we have $P = 648$ and $N = 22,201$ for the Urban data. Viewed this way, the problem under test may be viewed as matrix completion. Specifically, by measuring a small subset of the elements in each $\boldsymbol{x}_i$, selected uniformly at random, we equivalently have the problem of completing the matrix $\mathbf{X}$, with a small subset of observed entries, selected at random. Matrix completion is a subject of significant recent interest, with solutions for such providing an opportunity for comparison to the proposed solution.

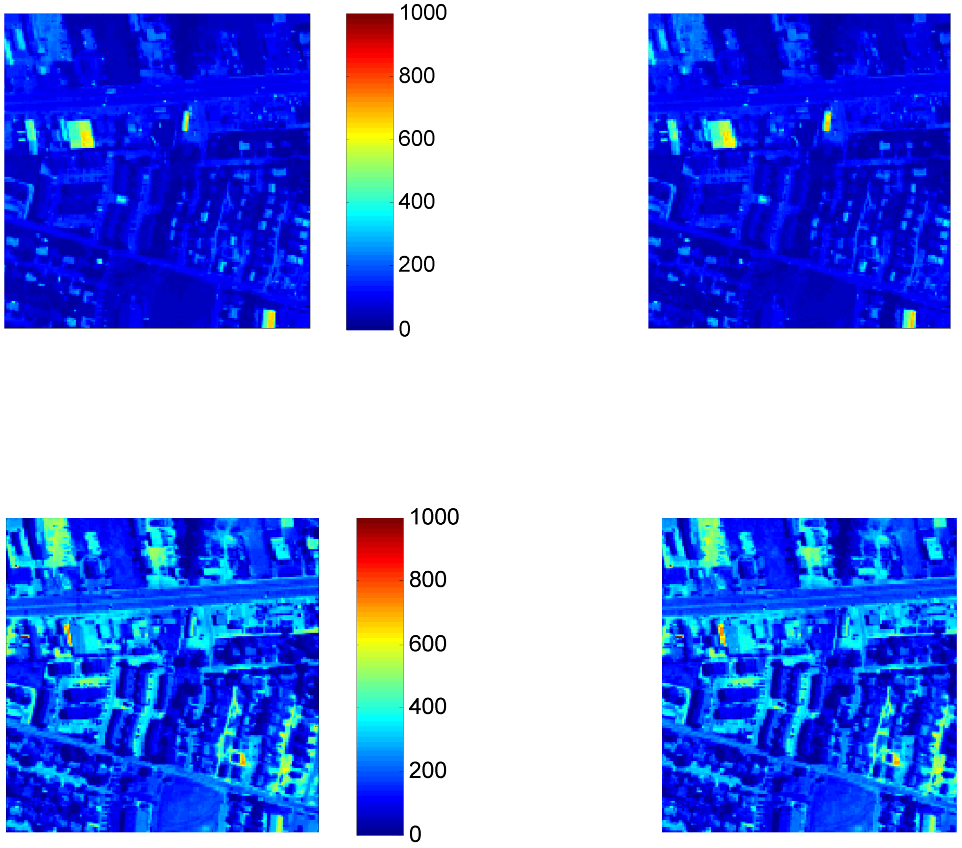

Fig. 6. Recovery of missing spectral bands from the Urban hyperspectral data. Of the 162 spectral bands, the data for 16 of the bands are removed entirely; of the remaining 146 bands, 5% of the voxels are sampled, selected uniformly at random. These figures present example recovery of the images at 2 of the 16 wavelengths for which data were missing entirely, based upon processing with 4×4 spatial blocks, and using the beta-process factor analysis model with a Gaussian process on the factor loadings. The left column corresponds to the original imagery at these two example wavelengths, and the right correspond to the recovered images (PSNR 38.3 dB for the recovered bands). The color scale on the right images is the same as that for the left.

To complete a matrix with missing data, we may assume that the original fully observed matrix satisfies a low-rank assumption, and use low-rank matrix completion algorithms [49], [50]. These methods minimize the ℓ_2 error between the observations and estimations, under the nuclear norm penalty. We test two state-of-the-art low-rank matrix completion algorithms, using Matlab code provided by the authors of these methods: the singular value thresholding (SVT) algorithm [49], and the augmented Lagrange multiplier (ALM) algorithm [50]. We find that even after careful parameter tuning, the SVT code[2] fails

[2] http://www-stat.stanford.edu/∼candes/svt/

TABLE II

ACCURACY OF RECOVERED DATACUBE (PSNR), BASED ON OBSERVING 2% AND 5% OF THE VOXELS, SELECTED UNIFORMLY AT RANDOM. RESULTS ARE SHOWN FOR THE URBAN DATA, CONSIDERING ANALYSIS OF 4×4 SPATIAL BLOCKS. RESULTS ARE SHOWN FOR THE BETA-PROCESS BASED FACTOR ANALYSIS (FA) MODEL (BPFA) AND FOR THE SHRINKAGE-BASED FA MODEL (SFA), IN EACH CASE WITH A GAUSSIAN PROCESS (GP) EMPLOYED FOR THE FACTOR LOADINGS. RESULTS ARE SHOWN FOR NOISE STANDARD DEVIATIONS OF 5, 15, 25, 35 AND 50, WHERE THE PSNR IS SHOWN, AS WELL AS THE INFERRED NOISE STANDARD DEVIATION. THE SAME NOISE STANDARD DEVIATION IS EMPLOYED AT ALL SPECTRAL BANDS. THE FIRST NUMBER IS THE INFERRED NOISE STANDARD DEVIATION, AND THE SECOND IS THE ASSOCIATED PSNR.

	5	15	25	35	50
GP-BPFA, 2%	10.4, 33.0	17.3, 31.9	26.4, 30.1	36.0, 29.9	51.2, 28.7
GP-SFA, 2%	8.5, 33.2	14.3, 31.9	21.8, 30.6	29.7, 29.5	41.5, 28.1
GP-BPFA, 5%	9.2, 39.1	16.8, 36.8	26.0, 34.9	35.7, 33.4	50.5, 31.8
GP-SFA, 5%	6.0, 40.1	14.3, 36.8	23.4, 34.6	32.8, 33.0	46.9, 31.3

to yield reasonable results for both HSI images considered in this paper. We present results of ALM[3] for the Urban data in Table IV. Although ALM works for both HSI images considered, it typically gives a rank-one estimation, which suggests that ALM is essentially substituting a weighted average for missing data. Thus it is not surprising that it does not provide good reconstruction and does not improve as either the spatial block size increases or the observed data ratio increases.

The KSVD algorithm exploits a related sparse representation for data reconstruction, and has demonstrated state-of-art results in gray-scale image restoration [9]. By introducing a weighed rank-one approximation and additional constraints to reduce color artifacts, KSVD has been extended to RGB color images (three spectral bands) [10]. We closely follow the extension from gray-scale to color images described in [10], to extent KSVD for HSI images. There are several parameters in KSVD that one must tune carefully, such as the sparsity level (the number of dictionary elements used by each spatial block vector) and the dictionary size. KSVD also requires good initialization, which we find crucial for HSI images. We initialized the KSVD dictionary elements by randomly sampling spatial blocks from the original complete HSI image, and using these as initial dictionary elements (this may be impossible in practice, and is only considered for comparison; random initialization of KSVD was ineffective for the

[3] http://perception.csl.uiuc.edu/matrix-rank/Files/inexact_alm_mc.zip

TABLE III

ACCURACY OF RECOVERED DATACUBE (PSNR), BASED ON OBSERVING 2% THROUGH 20% OF THE VOXELS, SELECTED UNIFORMLY AT RANDOM. RESULTS ARE SHOWN FOR THE URBAN DATA, CONSIDERING ANALYSIS WITH 2×2 AND 4×4 SPATIAL BLOCKS. RESULTS ARE SHOWN FOR THE BETA-PROCESS BASED FACTOR ANALYSIS (FA) MODEL (BPFA) AND FOR THE SHRINKAGE-BASED FA MODEL (SFA), IN EACH CASE WITH A GAUSSIAN PROCESS (GP) EMPLOYED FOR THE FACTOR LOADINGS. THE NOISE STANDARD DEVIATION AT EACH SPECTRAL BAND IS DRAWN FROM GAMMA$(75, 1/3)$.

	2×2, 5%	2×2, 10%	2×2, 15%	2×2, 20%	4×4, 5%	4×4, 10%	4×4, 15%	4×4, 20%
GP-BPFA	32.2	35.9	37.6	38.6	33.4	36.7	38.3	39.3
GP-SFA	32.8	35.8	37.3	38.3	34.7	37.4	38.9	39.9

TABLE IV

ACCURACY OF RECOVERED DATACUBE (PSNR), BASED ON OBSERVING 2% THROUGH 10% OF THE VOXELS, SELECTED UNIFORMLY AT RANDOM. RESULTS ARE SHOWN FOR THE URBAN DATA, CONSIDERING ANALYSIS WITH 2×2 AND 4×4 SPATIAL BLOCKS. RESULTS ARE SHOWN FOR THE AUGMENTED LAGRANGE MULTIPLIER (ALM) ALGORITHM AND FOR THE KSVD ALGORITHM.

	2×2, 2%	4×4, 2%	2×2, 5%	4×4, 5%	2×2, 10%	4×4, 10%
ALM	23.3	21.91	23.19	21.93	23.25	21.94
K-SVD	14.58	15.78	17.73	20.26	23.32	25.67

HSI data considered). The results of KSVD, after careful parameter tuning, are shown in Table IV. We find that KSVD improves as the spatial block sizes increases from 2×2 to 4×4 and the observed data ratio increases, but it performs much worse than the Bayesian algorithms considered (for which there has been no parameter tuning, and for which the dictionary was initialized at random).

Note that the above examples assumed no additive noise. The iterative methods are less attractive in the noisy case, as the noise level must be known to guide algorithm termination. Estimation of the noise level is difficult in the presence of substantial missing data, like that considered here. A significant advantage of the proposed approach is that the noise statistics are inferred jointly while learning the dictionary, and the noise statistics may vary with wavelength in the HSI data. It is very difficult to implement methods like SVT, ALM and KSVD in this setting.

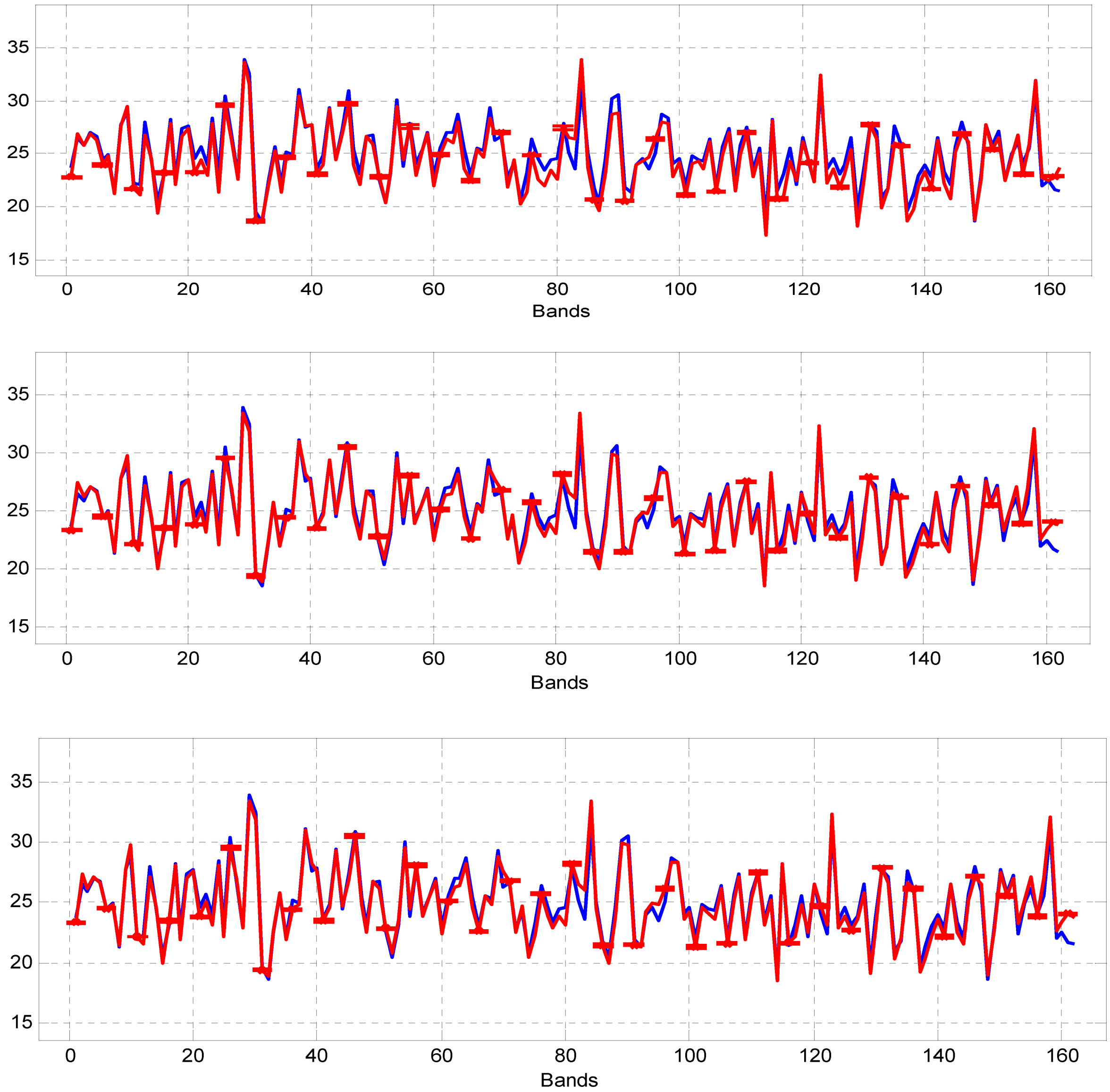

Fig. 7. True (blue) and estimated (red) noise variance, as a function of spectral band, using GP-BPFA. Results are shown for 4×4 spatial patches, and from top to bottom 10%, 15% and 20% of the voxels are observed, selected uniformly at random. Results are for the Urban hyperspectral data. The error bars on the inferred results correspond to one standard deviation, as computed from the posterior density function; only a subset of the error bars are shown, to enhance readability. The noise variance at each spectral band is drawn from Gamma$(75, 1/3)$.

V. Conclusions

Sparsity is playing an increasing role in many image processing problems [8]–[15]. In the analysis of hyperspectral imagery, researchers have used sparsity for endmember research [38]. In this paper we employed sparsity in a manner analogous to that utilized in previous endmember research, albeit here in a Bayesian manner. However, unlike in previous endmember studies, which were based on the spectral signature alone, in the analysis considered here the dictionary elements (analogous to endmembers) are learned while taking into account both spatial and spectral information.

A unique aspect of the work presented here is that rather than analyzing the entire datacube directly, we have processed a significantly downsampled version. Specifically, we have performed the analysis based on observing a small fraction of the voxels, selected uniformly at random. It was demonstrated that one may accurately recover the missing data, even in the presence of substantial wavelength-dependent noise.

There has been very little previous research in which the potential to massively down-sample a hyperspectral datacube has been considered. In addition, the manner in which that analysis has been performed here is also unique. Specifically, we have considered a fully Bayesian formulation, with previous techniques (applied to grey-scale or RGB imagery) based upon optimization approaches [8]–[14]. The Bayesian analysis yields “error bars” on all model parameters, of interest when one may desire a measure of confidence in the inferred missing data. Additionally, the Bayesian approach is well suited to analysis of noisy data, particularly when the noise statistics ($e.g.$, variance) is unknown and may be a function of wavelength.

In the context of denoising, particularly with wavelength-dependent noise variance, it has been found that imposition of smoothness on the factor loadings (as a function of wavelength) is critical to achieving accurate results. A Gaussian process [35] has been used to impose smoothness on the factor loadings, this having not been considered previously, for simpler image-processing tasks based on grey-scale or RGB imagery [29] (where the number of wavelengths is much less than that in hyperspectral data, and hence such smoothness constraints are unnecessary).

The Bayesian analysis has been performed using two constructions, one based upon use of shrinkage priors and the other based on a beta-Bernoulli construction. The former is related to previous research on Lasso [42], while the latter is related to the Indian buffet process [31]. The shrinkage construction has the advantage of close relationships with previous optimization-based approaches, with that linkage made explicit. It was found that the shrinkage and beta-Bernoulli approaches yielded similar results, with the latter much less sensitive to the truncation level on the number of factors. However, the beta-Bernoulli construction is significantly more efficient computationally, and therefore this is deemed to be the favored approach.

There are several directions of interest for future research. First, in all examples the analysis has been employed with no *a priori* training data. Specifically, learning of the dictionary and of the missing values has been performed only based upon the (significantly downsampled) data under test. While this is a good illustration of the power of the models, in practice one would expect to have available a database of potential signatures (not necessarily complete, but still providing useful prior information). It is of interest to combine such prior knowledge with the *in situ* dictionary-learning approach developed here. Imposition of such prior knowledge is anticipated to substantially improve modeling performance.

A second clear direction of future research concerns examination of material classification based upon hyperspectral datacubes recovered from massively downsampled measurements. This line of research is critical, as the principal objective of hyperspectral measurements concerns material characterization. Based upon the quality of the recovered data, as discussed in this paper, it is anticipated that high-quality material characterization will be achieved. Preliminary research in this direction is encouraging [51]. Finally, in this paper we have employed the GP to impose spectral smoothness in HSI data, but in imagery there are often other forms of structure that may be exploited [52]–[55]. Incorporation of such additional structure is likely to improve performance further.

REFERENCES

[1] B. Demir and S. Erturk, “Clustering based extraction of border training patterns for accurate SVM classification of hyperspectral images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 6, pp. 840–844, 2009.

[2] A. Zare and P. Gader, “Hyperspectral band selection and endmember detection using sparsity promoting priors,” *IEEE Geoscience and Remote Sensing Letters*, vol. 5, pp. 256–260, 2008.

[3] A. Zare, J. Bolton, P. Gader, and M. Schatten, “Vegetation mapping for landmine detection using long wave hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 172–178, 2008.

[4] J. Duarte-Carvajalino, G. Sapiro, M. Velez-Reyes, and P. Castillo, “Multiscale representation and segmentation of hyperspectral imagery using geometric partial differential equations and algebraic multigrid methods,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 2418–2434, 2008.

[5] A. Mohan, G. Sapiro, and E. Bosch, “Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images,” *IEEE Trans. Geosc. Remote Sensing Letters*, vol. 4, pp. 206–210, 2007.

[6] J. B. Lee, A. Woodyatt, and M. Berman, “Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform,” *IEEE Trans. Geosc. Remote Sensing*, vol. 28, p. 295304, 1990.

[7] A. Ifarraguerri and C.-I. Chang, “Unsupervised hyperspectral image analysis with projection pursuit,” *IEEE Trans. Geosc. Remote Sensing*, vol. 38, pp. 2529–2538, 2000.

[8] M. Aharon, M. Elad, and A. M. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Processing*, vol. 54, pp. 4311–4322, 2006.

[9] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Processing*, vol. 15, pp. 3736–3745, 2006.

[10] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Trans. Image Processing*, vol. 17, pp. 53–69, 2008.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proc. International Conference on Machine Learning*, 2009.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Supervised dictionary learning,” in *Proc. Neural Information Processing Systems*, 2008.

[13] J. Mairal, G. Sapiro, and M. Elad, “Learning multiscale sparse representations for image and video restoration,” *SIAM Multiscale Modeling and Simulation*, vol. 7, pp. 214 – 241, 2008.

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Proc. International Conference on Computer Vision*, 2009.

[15] M. Ranzato, C. Poultney, S. Chopra, and Y. Lecun, “Efficient learning of sparse representations with an energy-based model,” in *Proc. Neural Information Processing Systems*, 2006.

[16] J. Duarte-Carvajalino and G. Sapiro, “Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization,” *IEEE Transactions on Image Processing*, pp. 1395–1408, 2009.

[17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, “Self-taught learning: transfer learning from unlabeled data,” in *Proc. International Conference on Machine Learning*, 2007.

[18] A. Bruckstein, D. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM review*, vol. 51, pp. 34–81, 2007.

[19] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing*, 2009.

[20] E. Candès and T. Tao, “Near-optimal signal recovery from random projections: universal encoding strategies?” *IEEE Trans. Information Theory*, vol. 52, pp. 5406–5425, 2006.

[21] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[22] A. Charles, B. Olshausen, and C. Rozell, “Learning sparse codes for hyperspectral imagery,” *IEEE J. Selected Topics in Signal Processing*, 2011.

[23] R. Kawakami, J. Wright, Y.-W. Tai, Y. Matsushita, M. Ben-Ezra, and K. Ikeuchi, “High-resolution hyperspectral imaging via matrix factorization,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[24] J. Bobin, Y. Moudden, J.-L. Starck, and J. Fadili, “Sparsity constraints for hyperspectral data analysis: linear mixture model and beyond,” in *Proc. SPIE*, vol. 7446, 2009.

[25] Y. Moudden, J. Bobin, J.-L. Starck, and J. Fadili, “Dictionary learning with spatio-spectral sparsity constraints,” in *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2009.

[26] J. Greer, “Sparse demixing of hyperspectral images,” *IEEE Trans. Image Processing*, 2011.

[27] M. West, "Bayesian factor regression models in the "large p, small n" paradigm," *Bayesian Statistics*, vol. 7, pp. 723–732, 2003.

[28] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in *Proc. International Conference on Machine Learning*, 2009.

[29] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-parametric bayesian dictionary learning for sparse image representations," in *Proc. Neural Information Processing Systems*, 2009.

[30] R. Thibaux and M. Jordan, "Hierarchical beta processes and the indian buffet process," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2007.

[31] T. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *Proc. Advances in Neural Information Processing Systems*, 2005, pp. 475–482.

[32] D. Knowles and Z. Ghahramani, "Infinite sparse factor analysis and infinite independent components analysis," in *Proc. International Conference on Independent Component Analysis and Signal Separation*, 2007.

[33] M. Elad and I. Yavneh, "A weighted average of sparse representations is better than the sparsest one alone," *Preprint*, 2010.

[34] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, pp. 681–686, 2008.

[35] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[36] V. Paul, P. Piper, and R. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications*, vol. 416, pp. 29–47, 2006.

[37] W. Yin and S. Valiollahzadeh, "Hyperspectral data reconstruction combining spatial and spectral sparsity," *Rice University CAAM Technical Report TR10-29*, 2010.

[38] A. Zare and P. Gader, "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, p. 446450, 2007.

[39] M. Moller, E. Esser, S. Osher, G. Sapiro, and J. Xin, "A convex model for matrix factorization and dimensionality reduction on physical space and its application to blind hyperspectral unmixing," *UCLA CAM Report 10-71, http://www.math.ucla.edu/applied/cam/index.shtml*, 2010.

[40] A. Szlam, Z. Guo, and S. Osher, "A split bregman method for non-negative sparsity penalized least squares with applications to hyperspectral demixing," *UCLA CAM Report 10-06, http://www.math.ucla.edu/applied/cam/index.shtml*, 2010.

[41] J. Bernardo and A. Smith, *Bayesian theory*. Wiley, 2000.

[42] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[43] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, June 2001.

[44] H. Zou, "The adaptive lasso and its oracle properties," *J. Am. Stat. Ass.*, 2006.

[45] R. Gribonval, V. Cevher, and M. Davis, "Compressible distributions for high-dimensional statistics," *IEEE Trans. Information Theory*, submitted.

[46] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. Image Processing*, vol. 18, no. 6, pp. 1192–1202, 2009.

[47] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, pp. 97–109, 1970.

[48] V.Seshadri, *The Inverse Gaussian Distribution: a case study in exponential families*. Oxford Science Publications, 1993.

[49] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, 2010.

[50] Z. Lin, M. Chen, L. Wu, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *UIUC Technical Report UILU-ENG-09-2215*, 2009.

[51] A. Castrodad, Z. Xing, J. Greer, E. Bosch, L. Carin, and G. Sapiro, “Discriminative sparse representations in hyperspectral imagery,” in *Proc. Int. Conf. Image Proc. (ICIP)*, 2010.

[52] P. Garrigues and B. Olshausen, “Learning horizontal connections in a sparse coding model of natural images,” in *Advances in Neural Information Processing Systems*, 2008.

[53] V. Cevher, M. F. Duarte, C. Hedge, and R. G. Baraniuk, “Sparse signal recovery using Markov random fields,” in *Advances in Neural Information Processing Systems*, 2009.

[54] T. Faktor, Y. Eldar, and M. Elad, “Exploiting statistical dependencies in sparse representations for signal recovery,” *IEEE Trans. Signal Processing*, submitted.

[55] L. He and L. Carin, “Exploiting structure in wavelet-based Bayesian compressive sensing,” *IEEE Trans. Image Processing*, 2009.