Bayesian K-SVD Using Fast Variational Inference

Bayesian K-SVD Using Fast Variational Inference

Juan G. Serra*, Matteo Testa*, Rafael Molina, Aggelos K. Katsaggelos

***Abstract*—Recent work in signal processing in general and image processing in particular deals with sparse representation related problems. Two such problems are of paramount importance: an overriding need for designing a well-suited overcomplete dictionary containing a redundant set of atoms — i.e., basis signals— and how to find a sparse representation of a given signal with respect to the chosen dictionary. Dictionary learning techniques, among which we find the popular K-Singular Value Decomposition (K-SVD) algorithm, tackle these problems by adapting a dictionary to a set of training data. A common drawback of such techniques is the need for parameter-tuning. In order to overcome this limitation, we propose a fully-automated Bayesian method that takes into account the uncertainty of the estimates and produces a sparse representation of the data without prior information on the number of non-zeros in each representation vector. We follow a Bayesian approach that uses a three-tiered hierarchical prior to enforce sparsity on the representations and develop an efficient variational inference framework that reduces computational complexity. Furthermore, we describe a greedy approach that speeds up the whole process. Finally, we present experimental results that show superior performance on two different applications with real images: denoising and inpainting.**

***Index Terms*—Bayesian Modeling, Sparse Representation, K-SVD, Variational Inference, Dictionary Learning, Denoising, Inpainting.**

I. Introduction

Signal representation has drawn a lot of attention in the last decades. Be it a 1D signal, an image or a video, such representations should capture the most significant characteristics of the signal. These depend heavily on the application but seem to find a common goal in simplicity nonetheless.

Representing a signal requires the selection of a dictionary, i.e., a set of "atoms" or vectors in the signal space, a linear combination of which represents the given signal (alternative representations based on the use of manifolds [1], [2] are also relevant but will not be discussed here). The obvious and simplest choice of a dictionary is a basis, the smallest possible dictionary with the capability of representing the whole signal space. Simple as they are, the scarce expressiveness of such dictionaries led to the ongoing development of overcomplete dictionaries [3].

* Both authors contributed equally to this work.

M. Testa is with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin 10129, Italy (e-mail: matteo.testa@polito.it).

Juan G. Serra and R. Molina are with the Deptartmento de Ciencias de la Computación e I. A., Universidad de Granada, 18071 Granada, Spain (e-mail: jgserra@decsai.ugr.es; rms@decsai.ugr.es).

Aggelos K. Katsaggelos is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: aggk@eecs.northwestern.edu).

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under projects TIN2013-43880-R and DPI2016-77869-C2-2-R, by the Department of Energy grant DE-NA0002520 and by the European Research Council under the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant 279848.

The transition to overcomplete dictionaries was gradual. Analytical complete dictionaries were introduced first, which made use of different transforms such as DCT, Wavelet or Gabor. The limitations of such transforms were soon brought to light. Indeed, the work in [4] pointed out the deficiencies of the popular orthogonal wavelet transforms, namely its sensitivity to translation, dilation and rotation, resulting in the development of the Steerable Wavelet Transform. Early approaches towards overcomplete dictionaries tried to preserve the favorable orthogonality properties of bases but soon proved to be insufficient.

Parallel work suggested the use of collections of data to better describe signals, rather than the use of mathematical artificial functions. The works in [5] and [6] were very influential towards the recent advances in dictionary learning and sparse signal representation.

Let us now introduce the dictionary learning problem in a more formal way: we aim to find a sparse representation of each signal in a database of Q natural signals to $\mathbb{R}^P$ concatenated column-wise into a matrix as $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_Q] \in \mathbb{R}^{P \times Q}$. We do this by finding a set of K atoms in the signals' ambient space, concatenated into a dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_K] \in \mathbb{R}^{P \times K}$. This dictionary, and the corresponding assignment matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_Q] \in \mathbb{R}^{K \times Q}$ for the signals, are recovered by solving an optimization problem where we seek the best reconstruction of our signals given a budget T for the number of non-zero entries allowed in each column of $\mathbf{X}$. Formally this problem takes the form

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2 \\ &\text{s.t. } \|\mathbf{x}_q\|_0 \leq T, \;\; q = 1, \ldots, Q, \end{aligned} \tag{1}$$

where $\|\cdot\|_0$ denotes the ℓ_0-(pseudo)norm, which counts the non-zero entries in a vector, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Since the objective function $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2$ is not convex in $\mathbf{X}$ and $\mathbf{D}$ jointly, but biconvex, that is, convex in $\mathbf{X}$ and $\mathbf{D}$ individually, this problem can be addressed by alternating minimization over each variable separately. However, the exact minimization over $\mathbf{X}$ is well known to be NP-hard. Therefore greedy methods, among which the popular K-SVD algorithm [7], are used to approximate the true solution. Alternatively, the sparsity constraint can be relaxed, resulting in the following problem

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2 \\ &\text{s.t. } \|\mathbf{x}_q\|_1 \leq T, \;\; q = 1, \ldots, Q, \end{aligned} \tag{2}$$

where $\|\cdot\|_1$ denotes the vector ℓ_1-norm. A wide array of techniques from convex optimization can be applied to solve this problem (e.g., [7]–[10]).

The dictionary learning problem (in both forms of (1) and (2)) has been widely applied in image processing and machine learning. Applications include image denoising and deblurring [11], [12], image super-resolution [13], image restoration [14], face recognition [15], and classification [16] among many others. Focusing on this latter category, in [17] the authors propose to perform histopathological image classification through the use of class-specific dictionaries. In more detail, a class-specific dictionary should be able to correctly sparsify a signal belonging to the related class by employing just few atoms in the representation, but it should represent poorly a signal belonging to a different class. Thus, the classification is performed by analyzing the usage of atoms in the different dictionaries.

Back to the general dictionary learning problem, an alternative approach to the problem it is studied in the work of Skretting and Engan [18], referred to as the Recursive Least Squares Dictionary Learning Algorithm (RLS-DLA), according to which a continuous update of the dictionary is performed after each training vector is processed. Therein lies the main difference between RLS-DLA and other previous approaches such as its precedent ILS-DLA [19] or K-SVD [7]. However, its convergence has not been established. Liu et al. [20] propose a different method for the estimation of $\mathbf{D}$ and $\mathbf{X}$ using a two-level Bregman-based technique for MRI reconstruction. Its inner loop updates the sparse coefficients following an iterative shrinkage/thresholding algorithm, whereas the outer loop basically updates the atoms of the dictionary, which consists of a refinement of the previous one, involving just a matrix multiplication. A follow-up publication by the same authors, [21], further validates their previously presented method applying sparse representations to the image deconvolution problem.

For what concerns deterministic techniques for dictionary learning, another popular approach is the *analysis* approach. In contrast to the standard *synthesis* method in which the emphasis is placed on the signal decomposition by means of linear combinations of few atoms of a given dictionary, in the *analysis* counterpart the dictionary represents an operator which, multiplied by the signals, results in a sparse outcome. Examples of works following this approach include [22] and [23].

Along with deterministic methods to solve the dictionary learning problem, probabilistic approaches have also been proposed. In their seminal works, Olshausen and Field [6] and Lewicki and Sejnowski [24] introduced a generative model for the data which allowed them to develop a Maximum Likelihood (ML) estimator for both the sparse coding and the dictionary. According to this model, when the prior on the sparse signal is a heavily peaked Laplacian distribution around zero and the residual is approximated by a zero-mean Gaussian distribution, the dictionary learning problem reduces to the one in (2). Following this work, other authors proposed modifications to either the sparse approximation step, the dictionary update, or both. In [25], using the same generative model introduced in [6], the authors proposed the use of Orthogonal Matching Pursuit (OMP) to solve the sparse coding problem and a closed form solution for the dictionary update equation. Later papers focused on the use of a Maximum a Posteriori (MAP) approach instead, which allows to impose constraints on the dictionary as well. For instance, in the work of Kreutz-Delgado *et al.* [26] a unit-norm Frobenius prior is placed on the dictionary. However, due to the intractability of such a prior, they propose to use an approximate solution and the FOCUSS [27] algorithm in order to obtain the sparse solution. Other choices of priors involve smoother (less sparse) priors based on the Kullback-Leibler divergence for the ℓ_1 regularization as in [28]. The advantage of this latter approach lies in the increased stability of the sparse solution and the efficient convex inference.

All of the aforementioned techniques use ML or MAP estimators to solve the dictionary learning problem. However, the main drawback of such approaches is that they do not take into account the uncertainty of the estimated sparse representation coefficients, which, as we will later examine, leads to reduced algorithmic performance. Moreover, since the variance of the noise is not explicitly taken into account in the model, these algorithms have to rely on other techniques for noise estimation. The importance of a good estimate of the noise variance is discussed in [9] where the authors show that when using K-SVD for image denosing [11], the resulting PSNR is highly affected by the precision of the noise variance estimate.

To overcome these problems a few techniques have been developed. These include the incorporation of the noise variance/covariance information in the model as a parameter that can be estimated and taking into account the uncertainty of the estimates. The author in [29] propose an Expectation Maximization (EM) algorithm in which the posterior of the sparse signal is estimated along with the dictionary. In more detail, each column of $\mathbf{X}$ is modeled using a Laplacian prior which, however, leads to an intractable posterior distribution, for which the authors propose to use a variational approximation of the prior which tranforms the posterior of the sparse signal into a Gaussian form. Finally, an EM algorithm is developed in order to estimate the parameters of the model. However, with this approach, the authors do not place a prior on the entries of the dictionary.

Zhou *et al.*, [9] and [30], utilize a Beta-Bernoulli prior for the selection of the active-set, employing the Beta Process Factor Analysis (BPFA) modeling introduced in [31]. The active-set is the smallest possible set of atoms in the dictionary which is capable of efficiently explaining the underlying signal structure. Similarly to [32], the model can also be used to estimate the size of the dictionary. In addition, the authors introduced a Dirichlet patch clustering in order to group the data which have the same probability of being represented using a fixed set of atoms. Samples from the full posterior distribution are obtained through Gibbs sampling. BPFA modeling is also used in [33]. In this work, the problem of MRI image reconstruction from Compressed Sensing measurements is tackled with the introduction of a Total Variation (TV) penalty term in the functional, which is then solved through the use of the Alternating Direction Method of Multipliers (ADMM). The BPFA modeling falls into the category of the so-called *Spike and Slab* (SnS) prior models. The main

idea of this technique is to introduce a binary vector for the selection of the variables to be included in the model as in [34], where authors use this prior on the linear regression coefficients. This modeling is particularly useful when the number of observations is smaller than the number of possible predictors. The presence of the Dirac delta function in the model makes inference a difficult task. Authors have addressed this issue in several manners. Bayesian variational inference is used on SnS based models in [35], where a reparametrization of the SnS prior is proposed. The novelty of the work lies in a new factorization thanks to which each factor is a mixture of two Gaussians. The authors also show the effectiveness of such approach with respect to standard factorization as in [36], where the posterior approximation of the factors is a mixture of components with a single (unimodal) Gaussian distribution, which leads to a less accurate representation of the data. Based on this model, and motivated by the fact that posterior independence results in orthogonality of inferred atoms, [37] introduces an Expectation Maximization approach which does not assume independence among the dictionary elements. To efficiently deal with large datasets, the model in [30] was adapted to process randomly partitioned data in [38]. In more detail, the parameters of the model are inferred locally for each set of partitioned data and then aggregated using a weighted average to update the global parameters of the model resulting in an increased robustness to local minima and reduced memory requirements. Hansen *et al.* [39] propose a novel structured SnS prior which allows to incorporate prior knowledge on the support of the coefficient vectors via a Gaussian Process (GP) over the SnS probabilities; a later extension of their work, [40], covers the inference of the GP parameters. The GP models the sparsity patterns and considers correlations between the SnS probabilities of different coefficient vectors. In [41] the authors use a variant of SnS that replaces the Dirac delta function by an indicator function, based on the MAP estimation technique proposed in [42]. It allows to infer the optimal sparse coefficients by solving an optimization problem without relaxation. Finally, the work by Y. Zhang *et al.* [43] also utilizes the Bernoulli prior on the binary activations but imposes a multiplicative Gaussian process on the 2D coordinates of the image patches which enforces similarity on the support of neighboring patches. Furthermore, uncorrelation among dictionary atoms is favored by the use of a Sigmoid Belief Network.

Note that using variational inference on SnS priors, see for instance, [30] and [35], may not lead to an exact sparse solution since computing the expectation over a binary distribution will not necessarily produce a binary $\{0, 1\}$ value. However, this can be easily solved by using the thresholding approach in [38]. The method we propose in this work also leads to exact sparse solutions since it allows atoms to be added or removed completely from the model.

Additionally, we can find Dictionary Learning and Sparse Coding approximations which seek overall acceleration of the process. The works by Y. LeCun *et al.* use a multilayer feed forward network [44] or a binary tree [45] which make the algorithms suitable for real-time visual applications, such as object recognition. Along with these approximation techniques, [46] presents three different Dictionary Learning algorithms which also focus on computational efficiency. The authors propose partial updating of the atoms to accelerate convergence, a one-stage procedure in which each atom is updated along with its corresponding row in the coefficient matrix, letting the non-zero entries change, contrary to K-SVD, and lastly they incorporate the FISTA [47] sparse coding stage to the latter for faster performance.

Works which analyze the theoretical limits of the dictionary learning approach can also be found in the literature. In particular, in [48] the authors analyze the local minima of the non-convex functional in the dictionary learning problem. The results they obtain show that with high probability the sparse coding admits a local minimum around the dictionary which generated the signals. Additionally, C. Bao *et al.* [49] present a multi-block alternating proximal method with proven global convergence which is faster than K-SVD with similar performance.

In this work we propose a novel Bayesian algorithm for solving the ℓ_1 dictionary learning problems. Our approach aims at estimating the whole posterior distribution of $\mathbf{X}$ (thus taking into account the uncertainty of the estimated coefficients) but with an automatic technique for the estimation of the parameters which originate with the introduced models. The proposed approach is applied to image denoising and inpainting in order to test its performance in different applications of interest in image processing.

The paper is organized as follows. In Section II we briefly describe the K-SVD algorithm. Section III presents a hierarchical Bayesian model based on the use of the Laplace prior, and in Section IV we provide the details of the inference procedure to estimate the unknowns. Based on the inference procedure in Section IV, we develop a computationally efficient implementation based on Empirical Bayes in Section V-A. Numerical examples demonstrating the effectiveness of the proposed algorithm are given in Section VI, where we compare the results with state-of-the-art alternatives. Finally, we draw concluding remarks in Section VII.

Notation: Unless otherwise noted, throughout this paper, we use boldface upper-case and lower-case letters to denote matrices and vectors, respectively. For a matrix $\mathbf{X}$, its ith column and jth row are denoted by $\mathbf{x}_i$ and $\mathbf{x}^j$, respectively. The (i, j)-th entry of a matrix $\mathbf{X}$ is denoted by either x_{ij} or $\mathbf{X}(i, j)$, whichever makes the notation clearer. Given a vector $\mathbf{x}$, $\mathrm{diag}(\mathbf{x})$ represents the square matrix with the entries of $\mathbf{x}$ on its diagonal, while given a square matrix $\mathbf{X}$, $\mathrm{diag}(\mathbf{X})$ extracts its diagonal into a vector. Given a square matrix $\mathbf{X}$, $\mathrm{Tr}(\mathbf{X})$ and $|\mathbf{X}|$ denote the trace and determinant operators, respectively. The $M \times 1$ all-zero vector is denoted by $\mathbf{0}_M$, and finally, the $M \times M$ identity matrix is denoted by $\mathbf{I}_M$.

II. The K-SVD algorithm

Among the most popular algorithms for dictionary learning, K-SVD [7] is a greedy approach that approximately solves the standard ℓ_0 problem in (1). In K-SVD the optimization is performed coordinate-wise alternating between $\mathbf{X}$ and $\mathbf{D}$.

At each iteration of the K-SVD algorithm, given the current state update of the dictionary, the Orthogonal Matching Pursuit

(OMP) algorithm [50] is first applied to determine the support of $\mathbf{X}$, i.e., the locations of the non-zeros in $\mathbf{X}$, while the values at these non-zero locations obtained from OMP are discarded. Notice that this requires manually fixing the number of non-zero components in each column of $\mathbf{X}$.

After OMP, $\mathbf{D}$ and the non-zeros of $\mathbf{X}$ are updated. The term $\mathbf{DX}$ can be decomposed as

$$\mathbf{DX} = \sum_{k=1}^{K} \mathbf{d}_k \mathbf{x}^k \,. \tag{3}$$

This decomposition forms the basis of a cyclic update procedure, where each pair of $\{(\mathbf{d}_k, \mathbf{x}^k)\}_{k=1}^{K}$ is updated individually while all other pairs are held constant at their most recent values. Specifically, for the jth pair, the objective function in (1) can be expressed as the sum of a residual and a rank-one matrix, i.e.,

$$\min_{\mathbf{d}_j, \mathbf{x}^j} \|\mathbf{Y} - \mathbf{DX}\|_{\mathrm{F}}^2 = \min_{\mathbf{d}_j, \mathbf{x}^j} \|\mathbf{R}_j - \mathbf{d}_j \mathbf{x}^j\|_{\mathrm{F}}^2, \tag{4}$$

where the residual term

$$\mathbf{R}_j = \mathbf{Y} - \sum_{i \neq j} \mathbf{d}_i \mathbf{x}^i \tag{5}$$

does not depend on $(\mathbf{d}_j, \mathbf{x}^j)$. Because $\mathbf{d}_j \mathbf{x}^j$ has at most rank one, the minimization in (4) is precisely a low-rank approximation problem, which can be solved via the Singular Value Decomposition (SVD) [51].

Before computing the SVD of $\mathbf{R}_j$, we note that the support of $\mathbf{X}$ has already been determined using OMP. Resetting $\mathbf{x}^j$ via the SVD of $\mathbf{R}_j$ would destroy its sparse structure. To resolve this issue, instead of considering $\mathbf{R}_j$, we consider $\widetilde{\mathbf{R}}_j$, which is formed by retaining the columns of $\mathbf{R}_j$ that correspond to the non-zero entries in $\mathbf{x}^j$. As we will see later, this restricted processing has a clear justification in the Bayesian context. Concretely, we have

$$\mathbf{d}_j \widetilde{\mathbf{x}}^j = \sigma_1 \mathbf{u}_1 \mathbf{v}^1, \tag{6}$$

where σ_1 is the largest singular value of $\widetilde{\mathbf{R}}_j$, $\mathbf{u}_1$ and $\mathbf{v}^1$ are its corresponding left and right singular vectors, and $\widetilde{\mathbf{x}}^j$ denotes the row vector $\mathbf{x}^j$ after imposing the known sparsity support. After this step, the values at the non-zero locations of $\mathbf{x}^j$ are set equal to $\widetilde{\mathbf{x}}^j$. Notice that this restricted non-zero update does not have a mathematical justification and will reduce the quality of the SVD fitting. A justified way to alternate between atom and representation updates will be proposed in the coming sections.

The advantage of the K-SVD algorithm is its simplicity, as the update steps are greedy in nature. One major drawback, though, is that the uncertainty of the estimates of $\mathbf{D}$ and $\mathbf{X}$ is not taken into account in the estimation procedure. While not taking into account the uncertainty in the atoms of $\mathbf{D}$ may not be a problem due to the generally large number of columns in $\mathbf{X}$, each column of $\mathbf{X}$ normally has a reduced number of non-zero components and their inherent uncertainty should be accounted for. Furthermore, K-SVD requires to know the number of non-zero components in each column of $\mathbf{X}$, information that may not be available or may even be column dependent. In this paper we will show how these problems can be tackled in a principled manner using Bayesian modeling and inference.

III. Hierarchical Bayesian Model

A. Noise Modeling

The use of the sparsity inducing ℓ_1 norm in (2) requires an elaborate modeling. Following our previous work in [52], we begin by modeling the observation process by using

$$\mathrm{p}(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta) \propto \beta^{\frac{PQ}{2}} \exp\left\{-\frac{\beta}{2} \|\mathbf{Y} - \mathbf{DX}\|_{\mathrm{F}}^2\right\}, \tag{7}$$

where β is the noise precision. We assume that

$$\mathrm{p}(\beta|a^\beta, b^\beta) = \Gamma(\beta|a^\beta, b^\beta) \propto \beta^{a^\beta - 1} \exp(-b^\beta \beta), \tag{8}$$

with $a^\beta > 0$ and $b^\beta > 0$ being the shape and inverse scale parameters, respectively.

B. Modeling of $\mathbf{D}$ and $\mathbf{X}$

Since we expect the columns of $\mathbf{D}$ to be normalized vectors, we utilize the following prior on $\mathbf{D}$

$$\mathrm{p}(\mathbf{D}) = \prod_{k=1}^{K} \mathrm{p}(\mathbf{d}_k) \tag{9}$$

where

$$\mathrm{p}(\mathbf{d}_k) = \begin{cases} \text{const} & \text{if } \| \mathbf{d}_k \| = 1 \\ 0 & \text{elsewhere} \end{cases} \tag{10}$$

We now proceed to model the columns of $\mathbf{X}$. Although various general sparsity promoting priors could be considered here, see [53], we will only investigate the use of the Laplace prior on the components of the columns of $\mathbf{X}$ in this paper. The non-conjugacy of the likelihood in (7) and Laplace prior distributions makes the use of this prior for the columns of $\mathbf{X}$ intractable. In our approach we address this issue by applying instead a three-tiered hierarchical prior on each column of $\mathbf{X}$, which has the same sparsifying effect as a Laplace prior while rendering the inference tractable.

For each column $\mathbf{x}_q, q = 1, \ldots, Q$ of $\mathbf{X}$, we utilize

$$\begin{aligned} \mathrm{p}(\mathbf{x}_q|\boldsymbol{\gamma}_q) &= \prod_{k=1}^{K} \mathcal{N}(x_{kq}|0, \gamma_{kq}) \\ &= \mathcal{N}(\mathbf{x}_q|\mathbf{0}_K, \boldsymbol{\Gamma}_q) \end{aligned} \tag{11}$$

where $\boldsymbol{\gamma}_q$ is a $K \times 1$ column vector with elements γ_{kq}, $k = 1, \cdots, K$, and $\boldsymbol{\Gamma}_q = \mathrm{diag}(\boldsymbol{\gamma}_q)$ along with the tiered hyperpriors

$$\mathrm{p}(\boldsymbol{\gamma}_q|\lambda_q) = \prod_{k=1}^{K} \Gamma(\gamma_{kq}|1, \lambda_q/2) \tag{12}$$

and

$$\mathrm{p}(\lambda_q|\nu_q) = \Gamma(\lambda_q|\nu_q/2, \nu_q/2), \tag{13}$$

where we assume a flat distribution on ν_q.

With marginalization, this hierarchical model yields a Laplace distribution of $\mathbf{x}_q$ conditioned on λ_q

$$\mathrm{p}\left(\mathbf{x}_q|\lambda_q\right) = \frac{\lambda_q^{K/2}}{2^K} \exp\left\{-\sqrt{\lambda_q}\|\mathbf{x}_q\|_1\right\}. \tag{14}$$

C. Complete System Modeling

Throughout the remainder of this paper we will denote by

$$\boldsymbol{\Gamma} = \left[\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_Q\right],\ \boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_Q],\ \boldsymbol{\nu} = [\nu_1, \cdots, \nu_Q] \tag{15}$$

the hyperparameters associated with $\mathbf{X}$.

We also denote the entire set of unknowns as

$$\boldsymbol{\Theta} = \left\{ \{\mathbf{d}_k\}_{k=1}^K, \{\mathbf{x}_q\}_{q=1}^Q, \boldsymbol{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \beta \right\}. \tag{16}$$

Based on the above presented modeling, the complete system modeling is therefore given by the joint distribution

$$\mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) = \mathrm{p}(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta)\mathrm{p}(\beta)\mathrm{p}(\mathbf{D})\mathrm{p}(\mathbf{X}|\boldsymbol{\Gamma})\mathrm{p}(\boldsymbol{\Gamma}|\boldsymbol{\lambda})\mathrm{p}(\boldsymbol{\lambda}|\boldsymbol{\nu})\mathrm{p}(\boldsymbol{\nu}). \tag{17}$$

IV. Inference

Our scheme for estimating $\mathbf{D}$ and $\mathbf{X}$ depends on our ability to estimate the posterior distribution $\mathrm{p}(\boldsymbol{\Theta}|\mathbf{Y})$. We do this using variational distribution approximation [54]. Specifically, with Mean-Field Factorization, the joint posterior distribution is approximated as

$$\mathrm{q}(\boldsymbol{\Theta}) = \mathrm{q}(\beta)\mathrm{q}(\boldsymbol{\Gamma})\,\mathrm{q}(\boldsymbol{\lambda})\,\mathrm{q}(\boldsymbol{\nu})\mathrm{q}(\mathbf{X}) \prod_{k=1}^{K} \mathrm{q}(\mathbf{d}_k), \tag{18}$$

where in our case it is assumed that $\mathrm{q}(\boldsymbol{\Gamma})$, $\mathrm{q}(\boldsymbol{\lambda})$, and $\mathrm{q}(\boldsymbol{\nu})$ are degenerate distributions. We also assume that each $\mathrm{q}(\mathbf{d}_k), k = 1, \ldots, K$ is a degenerate distribution on a vector with $\| \mathbf{d}_k \|_2 = 1$.

For each $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$ where $\mathrm{q}(\boldsymbol{\theta}_i)$ is assumed to be degenerate, we can update its value by calculating

$$\hat{\boldsymbol{\theta}}_i = \arg\max_{\boldsymbol{\theta}_i} \ln \mathrm{q}(\boldsymbol{\theta}_i) = \arg\max_{\boldsymbol{\theta}_i} \langle \ln \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\boldsymbol{\theta}_i}, \tag{19}$$

where $\langle \cdot \rangle_{\boldsymbol{\Theta}\setminus\boldsymbol{\theta}_i}$ denotes the expectation taken with respect to all approximating factors $\mathrm{q}(\boldsymbol{\theta}_j)$, $j \neq i$.

For each $\boldsymbol{\theta}_i$ where $\mathrm{q}(\boldsymbol{\theta}_i)$ is assumed to be non-degenerate, we apply calculus of variations and obtain

$$\ln \mathrm{q}(\boldsymbol{\theta}_i) = \langle \ln\, \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\boldsymbol{\theta}_i} + C, \tag{20}$$

where C denotes a constant independent of the variable of current interest. For non-degenerate distributions $\mathrm{q}(\boldsymbol{\theta}_i)$, the updated value $\hat{\boldsymbol{\theta}}_i$ will denote its mean.

A. Estimation of $\mathbf{X}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\nu}$

In order to find an approximate posterior distribution of $\mathbf{X}$, we apply (20) and obtain

$$\begin{aligned}
\ln \mathrm{q}(\mathbf{X}) &= \langle \ln \mathrm{p}(\mathbf{Y}|\mathbf{D}, \mathbf{X}, \beta) + \ln \mathrm{p}(\mathbf{X}|\boldsymbol{\Gamma}) \rangle_{\boldsymbol{\Theta}\setminus\mathbf{X}} + C \\
&= \sum_{q=1}^{Q} \left\langle -\frac{\beta}{2} \|\mathbf{y}_q - \mathbf{D}\mathbf{x}_q\|_2^2 - \frac{1}{2}\mathbf{x}_q^{\mathrm{T}} \boldsymbol{\Gamma}_q^{-1} \mathbf{x}_q \right\rangle_{\boldsymbol{\Theta}\setminus\mathbf{x}_q} + C \\
&= \sum_{q=1}^{Q} \left\{ -\frac{\hat{\beta}}{2} \left\| \mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q \right\|_2^2 - \frac{1}{2}\mathbf{x}_q^{\mathrm{T}} \hat{\boldsymbol{\Gamma}}_q^{-1} \mathbf{x}_q \right\} + C.
\end{aligned} \tag{21}$$

It is clear from (21) that the columns of $\mathbf{X}$ in the posterior distribution approximation are independent with

$$\ln \mathrm{q}(\mathbf{x}_q) = -\frac{\hat{\beta}}{2} \|\mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q\|_2^2 - \frac{1}{2}\mathbf{x}_q^{\mathrm{T}} \hat{\boldsymbol{\Gamma}}_q^{-1} \mathbf{x}_q + C\,. \tag{22}$$

It is straightforward to see that $\mathrm{q}(\mathbf{x}_q)$ is a Gaussian distribution

$$\mathrm{q}(\mathbf{x}_q) = \mathcal{N}(\mathbf{x}_q | \hat{\mathbf{x}}_q, \boldsymbol{\Sigma}_{\mathbf{x}_q}) \tag{23}$$

with covariance matrix and mean vector defined respectively as

$$\boldsymbol{\Sigma}_{\mathbf{x}_q} = \left(\hat{\beta} \hat{\mathbf{D}}^{\mathrm{T}} \hat{\mathbf{D}} + \hat{\boldsymbol{\Gamma}}_q^{-1} \right)^{-1} \tag{24}$$

$$\hat{\mathbf{x}}_q = \hat{\beta} \boldsymbol{\Sigma}_{\mathbf{x}_q} \hat{\mathbf{D}}^{\mathrm{T}} \mathbf{y}_q. \tag{25}$$

Next, taking the appropriate expectation and finding a solution to (19) we can calculate the updates for the hyperparameters associated with $\mathbf{X}$

For $\boldsymbol{\gamma}_q$ we have the following optimization problem

$$\begin{aligned}
\hat{\boldsymbol{\gamma}}_q &= \arg\max_{\boldsymbol{\gamma}_q} \langle \ln \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\boldsymbol{\gamma}_q} \\
&= \arg\max_{\boldsymbol{\gamma}_q} \langle \ln \mathrm{p}\left(\mathbf{x}_q|\boldsymbol{\gamma}_q\right) \mathrm{p}\left(\boldsymbol{\gamma}_q|\lambda_q\right) \rangle_{\mathbf{x}_q, \lambda_q} + C,
\end{aligned} \tag{26}$$

where C contains all the terms which do not involve $\boldsymbol{\gamma}_q$. Using (11) and (12), we have

$$\begin{aligned}
\langle \ln \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\boldsymbol{\gamma}_q} = &-\frac{1}{2} \sum_{k=1}^{K} \log \gamma_{kq} - \frac{1}{2} \langle \mathbf{x}_q^T \boldsymbol{\Gamma}_q^{-1} \mathbf{x}_q \rangle_{\mathbf{x}_q} \\
&+ K \log \hat{\lambda}_q - \frac{\hat{\lambda}_q}{2} \sum_{k=1}^{K} \gamma_{kq} + C,
\end{aligned} \tag{27}$$

where

$$\langle \mathbf{x}_q^T \boldsymbol{\Gamma}_q^{-1} \mathbf{x}_q \rangle = \langle \mathbf{x}_q \rangle^T \boldsymbol{\Gamma}_q^{-1} \langle \mathbf{x}_q \rangle + \mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{x}_q} \boldsymbol{\Gamma}_q^{-1}). \tag{28}$$

We now find the optimal γ_{kq} by setting the derivative of the previous expression with respect to γ_{kq} equal to zero, which yields:

$$\hat{\gamma}_{kq} = -\frac{1}{2\hat{\lambda}_q} + \sqrt{\frac{1}{4\hat{\lambda}_q^2} + \frac{\hat{x}_{kq}^2 + \boldsymbol{\Sigma}_{\mathbf{x}_q}(k,k)}{\hat{\lambda}_q}}. \tag{29}$$

Following an analogous procedure, we have that

$$\begin{aligned}
\hat{\lambda}_q &= \arg\max_{\lambda_q} \langle \ln \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\lambda_q} \\
&= \arg\max_{\lambda_q} \langle \ln \mathrm{p}\left(\boldsymbol{\gamma}_q|\lambda_q\right) \mathrm{p}\left(\lambda_q|\nu_q\right) \rangle_{\boldsymbol{\gamma}_q, \nu_q} + C.
\end{aligned} \tag{30}$$

Again, expanding the previous expression using (12) and (13) we obtain

$$\langle \ln \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{\boldsymbol{\Theta}\setminus\lambda_q} = K \log \lambda_q - \frac{\lambda_q}{2} \sum_{k=1}^{K} \hat{\gamma}_{kq} \tag{31}$$

$$+ \left(\frac{\hat{\nu}_q}{2} - 1 \right) \log \lambda_q - \frac{\hat{\nu}_q}{2} \lambda_q + C, \tag{32}$$

and maximizing, the optimal $\hat{\lambda}_q$ is given by

$$\hat{\lambda}_q = \frac{\hat{\nu}_q + 2K - 2}{\hat{\nu}_q + \sum_{k=1}^{K} \hat{\gamma}_{kq}}\,. \tag{33}$$

Finally, for ν_q we have

$$\begin{aligned}\hat{\nu}_q &= \arg\max_{\nu_q} \langle \ln \mathrm{p}\left(\mathbf{Y}, \mathbf{\Theta}\right)\rangle_{\mathbf{\Theta}\setminus\nu_q} \\ &= \arg\max_{\nu_q} \langle \ln \mathrm{p}\left(\lambda_q|\nu_q\right) \mathrm{p}\rangle_{\lambda_q} + C,\end{aligned}$$

which, after using (13), produces

$$\frac{\nu_q}{2}\ln\frac{\nu_q}{2} - \ln\left(\Gamma\left(\frac{\nu_q}{2}\right)\right) + \frac{\nu_q}{2}\left(\ln\hat{\lambda}_q - \hat{\lambda}_q\right) . \tag{34}$$

This formula does not allow for an analytical solution, requiring numerical optimization to find the optimal $\hat{\nu}_q$.

B. Estimation of $\mathbf{D}$

First notice that we assume that the columns of $\mathbf{D}$ are independent of each other in the posterior distribution approximation, i.e.,

$$\mathrm{q}(\mathbf{D}) = \prod_{k=1}^{K} \mathrm{q}(\mathbf{d}_k) , \tag{35}$$

with these distributions degenerate on a point in $\| \mathbf{d}_k \| = 1$.

Focusing on a single $\mathbf{d}_k$ and applying (19), we have

$$\begin{aligned}&\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \left\langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2 \right\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} \\ &s.t. \quad \| \mathbf{d}_k \| = 1.\end{aligned} \tag{36}$$

We can write

$$\begin{aligned}&\langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = \langle \|\mathbf{Y} - \mathbf{D}\hat{\mathbf{X}} + \mathbf{D}(\hat{\mathbf{X}} - \mathbf{X})\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} \\ &= \langle \|\mathbf{Y} - \mathbf{D}\hat{\mathbf{X}}\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} + \langle \|\mathbf{D}(\hat{\mathbf{X}} - \mathbf{X})\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k}\end{aligned} \tag{37}$$

where the cross terms are not included since both are identical and equal to zero. For the first term we have

$$\begin{aligned}\langle \|\mathbf{Y} - \mathbf{D}\hat{\mathbf{X}}\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} &= \|\mathbf{Y} - \sum\nolimits_{i\neq k} \hat{\mathbf{d}}_i\hat{\mathbf{x}}^i - \mathbf{d}_k\hat{\mathbf{x}}^k\|_{\mathrm{F}}^2 \\ &= \|\hat{\mathbf{x}}^k\|_2^2\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k - 2\mathbf{b}_k^{\mathrm{T}}\mathbf{d}_k + C\end{aligned} \tag{38}$$

with

$$\mathbf{b}_k = (\mathbf{Y} - \sum\nolimits_{i\neq k} \hat{\mathbf{d}}_i\hat{\mathbf{x}}^i)(\hat{\mathbf{x}}^k)^{\mathrm{T}}. \tag{39}$$

Notice that (38) is the only term used in K-SVD to update the atoms of the dictionary.

The uncertainty of the estimate of $\mathbf{x}_q$ is incorporated in the estimation of $\mathbf{d}_k$ by the second term on the right hand side of (37) which we now calculate. It can be expressed as

$$\begin{aligned}&\langle \|\mathbf{D}(\hat{\mathbf{X}} - \mathbf{X})\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = \langle \|\mathbf{d}_k(\hat{\mathbf{x}}^k - \mathbf{x}^k)\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} \\ &+ 2\langle \mathrm{Tr}(\mathbf{d}_k(\hat{\mathbf{x}}^k - \mathbf{x}^k)(\sum_{i\neq k} \hat{\mathbf{d}}_i(\hat{\mathbf{x}}^i - \mathbf{x}^i))^{\mathrm{T}})\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} + C.\end{aligned} \tag{40}$$

Now, the first term on the right hand side of (40) can be written as

$$\langle \|\mathbf{d}_k(\hat{\mathbf{x}}^k - \mathbf{x}^k)\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = c_k\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k, \tag{41}$$

where

$$\begin{aligned}c_k &= \langle \|\hat{\mathbf{x}}^k - \mathbf{x}^k\|_2^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = \langle \sum_{q=1}^{Q} (\hat{x}_{kq} - x_{kq})^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} \\ &= \sum_{q=1}^{Q} \mathbf{\Sigma}_{\mathbf{x}_q}(k,k),\end{aligned} \tag{42}$$

and $\mathbf{\Sigma}_{\mathbf{x}_q}(k,k)$ denotes the (k,k)-th element of $\mathbf{\Sigma}_{\mathbf{x}_q}$ defined in (24).

Similarly, the second term on the right hand side of (40) can be written as

$$\begin{aligned}&\langle \mathrm{Tr}(\mathbf{d}_k(\hat{\mathbf{x}}^k - \mathbf{x}^k)(\sum_{i\neq k} \mathbf{d}_i(\hat{\mathbf{x}}^i - \mathbf{x}^i))^{\mathrm{T}})\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} \\ &= \langle (\hat{\mathbf{x}}^k - \mathbf{x}^k)(\sum_{i\neq k} (\hat{\mathbf{x}}^i - \mathbf{x}^i)^{\mathrm{T}}\mathbf{d}_i^{\mathrm{T}})\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k}\mathbf{d}_k \\ &= \sum_{i\neq k} \langle (\hat{\mathbf{x}}^k - \mathbf{x}^k)(\hat{\mathbf{x}}^i - \mathbf{x}^i)^{\mathrm{T}}\mathbf{d}_i^{\mathrm{T}}\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k}\mathbf{d}_k = \mathbf{a}_k^{\mathrm{T}}\mathbf{d}_k,\end{aligned} \tag{43}$$

where

$$\mathbf{a}_k = \sum_{q=1}^{Q}\sum_{i\neq k} \mathbf{\Sigma}_{\mathbf{x}_q}(i,k)\hat{\mathbf{d}}_i . \tag{44}$$

Substituting (41) and (43) into (40), we obtain

$$\langle \|\mathbf{D}(\hat{\mathbf{X}} - \mathbf{X})\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = c_k\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k + 2\mathbf{a}_k^{\mathrm{T}}\mathbf{d}_k + C, \tag{45}$$

and substituting (38) and (45) into (37), we obtain

$$\begin{aligned}&\langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = e_k\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k - 2(\mathbf{b}_k - \mathbf{a}_k)^{\mathrm{T}}\mathbf{d}_k + C \\ &= \|\sqrt{e_k}\mathbf{d}_k - \frac{1}{\sqrt{e_k}}(\mathbf{b}_k - \mathbf{a}_k)\|^2 + C,\end{aligned} \tag{46}$$

where

$$e_k = \|\hat{\mathbf{x}}^k\|^2 + c_k . \tag{47}$$

Defining

$$\mathbf{t}_k = \frac{1}{\sqrt{e_k}}(\mathbf{b}_k - \mathbf{a}_k) \tag{48}$$

we obtain

$$\langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2\rangle_{\mathbf{\Theta}\setminus\mathbf{d}_k} = \|\mathbf{t}_k - \sqrt{e_k}\mathbf{d}_k\|^2 + C . \tag{49}$$

We can therefore finally write

$$\begin{aligned}&\hat{\mathbf{d}}_k = \arg\min \|\mathbf{t}_k - \sqrt{e_k}\mathbf{d}_k\|^2 \\ &\quad s.t. \ \| \mathbf{d}_k \|^2 = 1,\end{aligned} \tag{50}$$

which produces

$$\hat{\mathbf{d}}_k = \frac{1}{\| \mathbf{t}_k \|}\mathbf{t}_k = \frac{\mathbf{b}_k - \mathbf{a}_k}{||\mathbf{b}_k - \mathbf{a}_k||}. \tag{51}$$

C. Estimation of Noise Precision β

Keeping the terms dependent on β in (17) and applying (20), we obtain

$$\begin{aligned}\ln \mathrm{q}(\beta) =& \frac{PQ}{2}\ln\beta - \frac{\beta}{2}\left\langle \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathrm{F}}^2\right\rangle_{\mathbf{\Theta}\setminus\beta} \\ &+ (a^\beta - 1)\ln\beta - b^\beta\beta + C,\end{aligned} \tag{52}$$

from which we see that $\mathrm{q}(\beta)$ is a Gamma distribution with mean

$$\hat{\beta} = \frac{PQ + 2a^\beta}{\sum_{q=1}^{Q}\langle \|\mathbf{y}_q - \hat{\mathbf{D}}\mathbf{x}_q\|^2\rangle_{\mathbf{x}_q} + 2b^\beta} . \tag{53}$$

V. Fast Inference Procedure Based on Empirical Bayes

The inference procedure introduced in the previous section is mathematically sound but it can be computationally challenging and memory intensive since computing $\mathbf{\Sigma}_{\mathbf{x}_q}$ in (24) for each q requires the inversion of a $K \times K$ matrix at each iteration step.

In order to reduce the computational complexity and alleviate memory requirements, we propose a fast inference procedure based on the use of Empirical Bayes [52], [55], [56]. The principle of this approach is first presented in [55] in the context of Sparse Bayesian Learning (SBL) and later adapted in [52] and [56] for recovery of sparse signals. Here we adapt it for the sparse dictionary learning problem.

Specifically, for each $\mathbf{x}_q$, we adopt a constructive approach for identifying its support, i.e., the locations where it assumes non-zero values. The values of the hyperparameters at these non-zero locations are obtained via Maximum *A Posteriori* (MAP) estimation. With such support identification and hyperparameter estimation, the effective problem dimensions are drastically reduced due to sparsity. Finally, the estimated values of $\mathbf{x}_q$ in its support are obtained via (25).

A. Fast Bayesian Inference for $\mathbf{\Gamma}$ and $\mathbf{X}$

We will derive in this section a fast inference approach for $\mathbf{\Gamma}$ and $\mathbf{X}$. We start from the observation model

$$\mathbf{y}_q = \mathbf{D}\mathbf{x}_q + \mathrm{n}_q, \tag{54}$$

and the prior on $\mathbf{x}_q$ given by (11). Then, using $\hat{\mathbf{D}}$ and $\hat{\beta}$ and integrating on $\mathbf{x}_q$, we have

$$\mathrm{p}(\mathbf{y}_q|\hat{\beta}, \hat{\mathbf{D}}, \boldsymbol{\gamma}_q) = \mathcal{N}(\mathbf{y}_q|\mathbf{0}_P, \mathbf{C}_q), \tag{55}$$

where

$$\mathbf{C}_q = \hat{\beta}^{-1}\mathbf{I}_P + \hat{\mathbf{D}}\mathbf{\Gamma}_q\hat{\mathbf{D}}^{\mathrm{T}}. \tag{56}$$

We can then write

$$\begin{aligned}&\log \mathrm{p}(\mathbf{y}_q|\hat{\beta}, \hat{\mathbf{D}}, \boldsymbol{\gamma}_q)\mathrm{p}(\boldsymbol{\gamma}_q|\hat{\lambda}_q)\\ &= -\frac{1}{2}\Big[\log|\mathbf{C}_q| + \mathbf{y}_q^{\mathrm{T}}{\mathbf{C}_q}^{-1}\mathbf{y}_q + \hat{\lambda}_q\sum_k \gamma_{kq}\Big] + C,\end{aligned} \tag{57}$$

where C contains all terms which do not depend on $\boldsymbol{\gamma}_q$. We now replace the previously described EM procedure to estimate $\boldsymbol{\gamma}_q$ (and the the posterior distribution of $\mathbf{x}_q$) by the direct maximization of

$$\mathcal{L}(\boldsymbol{\gamma}_q) = -\frac{1}{2}\Big[\log|\mathbf{C}_q| + \mathbf{y}_q^{\mathrm{T}}{\mathbf{C}_q}^{-1}\mathbf{y}_q + \hat{\lambda}_q\sum_k \gamma_{kq}\Big]. \tag{58}$$

Notice that once $\hat{\boldsymbol{\gamma}}_q$ has been calculated we can easily find the posterior distribution of $\mathbf{x}_q$. Furthermore, if $\boldsymbol{\gamma}_{kq} = 0$, then the posterior distribution of x_{kq} will be degenerate at zero.

Let us examine how to add, update (or remove) a single γ_{kq} in order to increase $\mathcal{L}(\boldsymbol{\gamma}_q)$. Observing (56) we see that we can separate the contribution of a single γ_{kq} in $\mathbf{C}_q$ and write

$$\begin{aligned}\mathbf{C}_q &= \hat{\beta}^{-1}\mathbf{I}_P + \sum_{i\neq k}\gamma_{iq}\hat{\mathbf{d}}_i\hat{\mathbf{d}}_i^{\mathrm{T}} + \gamma_{kq}\hat{\mathbf{d}}_k\hat{\mathbf{d}}_k^{\mathrm{T}}\\ &\stackrel{\text{def}}{=} {}^{-k}\mathbf{C}_q + \gamma_{kq}\hat{\mathbf{d}}_k\hat{\mathbf{d}}_k^{\mathrm{T}},\end{aligned} \tag{59}$$

where, clearly, ${}^{-k}\mathbf{C}_q$ denotes the terms not including γ_{kq}.

Using the matrix inversion lemma and the determinant identity on $\mathbf{C}_q$ we obtain

$$\mathbf{C}_q^{-1} = {}^{-k}\mathbf{C}_q^{-1} - \frac{{}^{-k}\mathbf{C}_q^{-1}\hat{\mathbf{d}}_k\hat{\mathbf{d}}_k^{\mathrm{T}}\,{}^{-k}\mathbf{C}_q{}^{-1}}{\gamma_{kq}^{-1} + \hat{\mathbf{d}}_k^{\mathrm{T}}{}^{-k}\mathbf{C}_q^{-1}\hat{\mathbf{d}}_k}, \tag{60}$$

$$|\mathbf{C}_q| = |{}^{-k}\mathbf{C}_q||1 + \gamma_{kq}\hat{\mathbf{d}}_k^{\mathrm{T}}\,{}^{-k}\mathbf{C}_q{}^{-1}\hat{\mathbf{d}}_k|\,. \tag{61}$$

These two equations allow us to rewrite (58) as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}_q) &= -\frac{1}{2}\Big[\log|{}^{-k}\mathbf{C}_q| + \mathbf{y}_q^{\mathrm{T}}{}^{-k}\mathbf{C}_q^{-1}\mathbf{y}_q + \hat{\lambda}_q\sum_{n\neq k}\gamma_{nq}\Big]\\ &\quad + \frac{1}{2}\Big[\log\frac{1}{1+\gamma_{kq}s_{iq}} + \frac{h_{kq}^2\gamma_{kq}}{1+\gamma_{kq}s_{kq}} - \hat{\lambda}_q\gamma_{iq}\Big]\\ &= \mathcal{L}({}^{-k}\boldsymbol{\gamma}_q) + l(\gamma_{kq}),\end{aligned} \tag{62}$$

where

$$l(\gamma_{kq}) = \frac{1}{2}\left[\log\frac{1}{1+\gamma_{kq}s_{kq}} + \frac{h_{kq}^2\gamma_{kq}}{1+\gamma_{kq}s_{kq}} - \lambda_q\gamma_{kq}\right] \tag{63}$$

and s_{kq} and h_{kq} are defined as

$$\begin{aligned}s_{kq} &= \hat{\mathbf{d}}_k^{\mathrm{T}}{}^{-k}\mathbf{C}_q^{-1}\hat{\mathbf{d}}_k\\ h_{kq} &= \hat{\mathbf{d}}_k^{\mathrm{T}}{}^{-k}\mathbf{C}_q^{-1}\mathbf{y}_q\end{aligned}\,. \tag{64}$$

The quantities s_{kq} and h_{kq} do not depend on γ_{kq}. Therefore, the terms related to a single hyperparameter γ_{kq} are now separated from the rest. A closed form solution of the maximization of $\mathcal{L}(\boldsymbol{\gamma}_q)$, when only its kth component is changed, can be found by holding the other hyperparameters fixed, taking its derivative with respect to γ_{kq} and setting it equal to zero.

The optimal $\hat{\gamma}_{kq}$ can be obtained as follows (see [52] for details)

$$\hat{\gamma}_{kq} = m_{kq}\mathbf{1}_{[h_{kq}^2 - s_{kq} \geq \hat{\lambda}_q]} \tag{65}$$

where

$$\begin{aligned}m_{kq} &= -\frac{s_{kq}(s_{kq}+2\hat{\lambda}_q)}{2\hat{\lambda}_q s_{kq}^2}\\ &\quad + \frac{s_{kq}\sqrt{(s_{kq}+2\hat{\lambda}_q)^2 - 4\hat{\lambda}_q(s_{kq} - h_{kq}^2 + \hat{\lambda}_q)}}{2\hat{\lambda}_q s_{kq}^2}\end{aligned} \tag{66}$$

It is crucial to perform all the calculations efficiently. To explain how they can be carried out we overload the notation slightly. We rewrite the current (^{c}) covariance matrix of the marginal of the observations as

$$\mathbf{C}_q^c = \hat{\beta}^{-1}\mathbf{I}_P + \sum_{i\in\mathcal{A}}\gamma_{iq}^c\hat{\mathbf{d}}_i\hat{\mathbf{d}}_i^{\mathrm{T}} + \sum_{i\in\overline{\mathcal{A}}}\gamma_{iq}^c\hat{\mathbf{d}}_i\hat{\mathbf{d}}_i^{\mathrm{T}}, \tag{67}$$

where $\mathcal{A} = \{i|\gamma_{iq}^c > 0\}$ and $\overline{\mathcal{A}} = \{i|\gamma_{iq}^c = 0\}$. The last term on the right hand side of the above equation is zero and has been included for clarity.

Then using the Woodbury identity we have

$$\hat{\mathbf{d}}_k^{\mathrm{T}}\mathbf{C}_q^{c\,-1}\hat{\mathbf{d}}_k = \hat{\beta}\hat{\mathbf{d}}_k^{\mathrm{T}}\hat{\mathbf{d}}_k - \hat{\beta}^2\hat{\mathbf{d}}_k^{\mathrm{T}}\hat{\mathbf{D}}^c\Sigma_{\mathbf{x}_q}^c(\hat{\mathbf{D}}^c)^{\mathrm{T}}\hat{\mathbf{d}}_k \stackrel{\text{def}}{=} S_{kq} \tag{68}$$

where $\Sigma_{\mathbf{x}_q}^c$ is obtained from $\Sigma_{\mathbf{x}_q}$ by keeping only the columns and rows associated to the indices in $\mathcal{A}$. The same restriction

applies to the columns of $\hat{\mathbf{D}}^c$, that is, we keep in $\hat{\mathbf{D}}^c$ the columns associated to $\gamma_{iq}^c > 0$.

From (60) we obtain, for $k \in \mathcal{A} \cup \overline{\mathcal{A}}$,

$$s_{kq} = \frac{S_{kq}}{1 - \gamma_{kq}^c S_{kq}}. \tag{69}$$

Furthermore

$$\hat{\mathbf{d}}_k^{\mathrm{T}} \mathbf{C}_q^{c\,-1} \mathbf{y}_q = \hat{\beta} \hat{\mathbf{d}}_k^{\mathrm{T}} \mathbf{y}_q - \hat{\beta}^2 \hat{\mathbf{d}}_k^{\mathrm{T}} \hat{\mathbf{D}}^c \Sigma_{\mathbf{x}_q}^c (\hat{\mathbf{D}}^c)^{\mathrm{T}} \mathbf{y}_q \stackrel{\text{def}}{=} Q_{kq} \tag{70}$$

and following the same procedure we obtain

$$q_{kq} = \frac{Q_{kq}}{1 - \gamma_{kq}^c S_{kq}}. \tag{71}$$

Given $\Sigma_{\mathbf{x}_q}^c$ we now have an efficient procedure to check whether we should add $\gamma_{kq}, k \in \overline{\mathcal{A}}$, or update (or remove) $\gamma_{kq}, k \in \mathcal{A}$. Furthermore, the amount the marginal log likelihood is improved by each single addition, update (or removal) is easily obtained from (62). Finally we notice that $\boldsymbol{\Sigma}_q^c$ and $\hat{\mathbf{x}}_q^c$ can be updated very efficiently when only a single coefficient γ_{kq} is considered, see [55].

With the fast updates just described in hand we can formally state the full **Algorithm 1** which we henceforth refer to as the Bayesian K-SVD (BKSVD) method.

Notice that at step 7 of the algorithm, a candidate γ_{kq} must be selected for updating. This can be done by randomly choosing a basis vector $\hat{\mathbf{d}}_k$, or by calculating each γ_{kq} and choosing the one that results in the greatest increase in $\mathcal{L}(\boldsymbol{\gamma}_q)$, which results in a faster convergence. The latter is the method implemented in this work.

An important contribution of the algorithm is the estimation of the noise precision β, which is derived in the previous section using (53).

In the approach presented in [57], the estimation of the noise precision is unreliable at early iterations at which it is necessary to set it to a fixed value. Unreliable estimates can indeed significantly affect the performance of the technique. However, in the proposed method β is estimated using a set of signals which are assumed to share the same noise variance, thus leading to reliable estimates even at early BKSVD iterations.

For a comparison of the proposed fast algorithm with the Relevance Vector Machine (RVM) when the dictionary is fixed, refer to [52].

We now relate the proposed BKSVD model to K-SVD. In K-SVD the number of non-zero components, S, in $\mathbf{x}_q$ is fixed in advance. In BKSVD we can update $\boldsymbol{\gamma}_q$ until convergence and then keep only its S largest values. We can also run BKSVD in a greedy fashion until S non-zero components are incorporated.

Finally, let us compare the iteration procedures for BKSVD and K-SVD. In K-SVD to update the kth atom we select the non-zero components in $\hat{\mathbf{x}}^k$. If the qth component is selected, this means that γ_{qk} is non-zero in our fast formulation. Notice that the components selected by BKSVD ($\gamma_{qk} \neq 0$) and the ones selected by K-SVD ($\hat{x}_{qk} \neq 0$) coincide almost surely. K-SVD then proceeds to find the rank-one SVD decomposition of the residual term

$$\mathbf{R}_k = \mathbf{Y} - \sum_{i \neq k} \hat{\mathbf{d}}_i \hat{\mathbf{x}}^i \tag{72}$$

where only the columns $\mathbf{y}_q$ with non-zero γ_{kq} are considered. This produces an update of $\mathbf{d}_k$ and the non-zero components of $\hat{\mathbf{x}}^k$. On the other hand, BKSVD not only takes into account the residual $\mathbf{R}_k$ in $\|\mathbf{Y} - \sum_{i \neq k} \hat{\mathbf{d}}_i \hat{\mathbf{x}}^i - \mathbf{d}_k \hat{\mathbf{x}}^k\|_{\mathrm{F}}^2$, see (38), but also makes the uncertainty of the estimation of $\mathbf{X}$ reponsible for some of the variation of the model, see $\langle \|\mathbf{D}(\hat{\mathbf{X}} - \mathbf{X})\|_{\mathrm{F}}^2 \rangle_{\boldsymbol{\Theta} \setminus \mathbf{d}_k}$ in (40).

To update the kth atom BKSVD utilizes (51), while K-SVD utilizes the rank-one decomposition of $\mathbf{R}_k$ to update $\hat{\mathbf{d}}_k$ and the non-zero elements in $\mathbf{x}^k$. For BKSVD, once the kth atom has been updated we can also update the non-zero components in $\mathbf{x}^k$. Both strategies will be compared in the experimental section.

Algorithm 1 Pseudocode for BKSVD algorithm

1: Input: $\mathbf{Y}$, initial normalized $\mathbf{D}$
2: Output: $\hat{\mathbf{D}}$, $\hat{\boldsymbol{\Gamma}}$, the posterior approximations $\mathrm{q}(\mathbf{x}_q)$, $q = 1, \ldots, Q$
3: initialize $\boldsymbol{\Gamma}$ and $\boldsymbol{\lambda}$ to zero
4: **while** not converged **do**
5: **for** q in $1, \ldots, Q$ **do**
6: **while** not converged **do**
7: Choose a $k \in \{1, \ldots, K\}$ (or equivalently choose a γ_{kj})
8: Find optimal $\hat{\gamma}_{kq}$ using (65)
9: Update $\boldsymbol{\Sigma}_{\mathbf{x}_q}$ and $\hat{\mathbf{x}}_q$ based on $\hat{\gamma}_{kq}$
10: Update s_{kq} and h_{kq}
11: Update $\hat{\lambda}_j$ and $\hat{\nu}_j$
12: **end while**
13: **end for**
14: **for** k in $1, \ldots, K$ **do**
15: Update $\mathbf{d}_k$ using (51)
16: **end for**
17: Update $\hat{\beta}$
18: **end while**

We finally provide in this section a discussion on how our method relates to existing spike and slab approaches. We concentrate on $\mathbf{x}_q$ modeling since the dictionary updates can be considered to be similar. First we note that a clear explanation of the spike and slab prior modeling can be found in [58] where each $\mathbf{x}_q$ in our model is replaced by the Hadamard product $\mathbf{s}_q \odot \mathbf{w}_q$ with $\mathrm{p}(\mathbf{s}_q) = \prod_{k=1}^K \mathrm{Ber}(s_{kq}|\pi_k)$, where π_k is the probability of the kth atom to appear in the sparse representation of $\mathbf{y}_q$ and $\mathrm{p}(\mathbf{w}_q) = \mathcal{N}(\mathbf{w}_q|\mathbf{0}, \gamma_s^{-1}\mathbf{I})$. The prior distribution on $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^{\mathrm{T}}$ is assumed to be $p(\boldsymbol{\pi}) = \prod_{k=1}^K \mathrm{Beta}(a/K, b(K-1)/K)$.

As explained in [30], as $K \to \infty$, and integrating on $\boldsymbol{\pi}$, the draws on $\{\mathbf{s}_q\}$ should be sparse and there should be a relatively consistent (re)use of dictionary elements across all $\mathbf{y}_q$, thereby also imposing self-similarity. This reuse of atoms, which can be an interesting property, can also lead to a lower incoherence of the dictionary as it will be reported in the

experimental section. On the other hand, the use of the same $\boldsymbol{\pi}$ realization for all $\mathbf{s}_q$ and also the same precision γ_s^{-1} for all $\mathbf{w}_q$ leads to fewer parameters to be estimated and also to reduced overfitting (note also the properties of the Beta process used). Obviously $\boldsymbol{\pi}$ and γ_s could be sampled independently for each $\mathbf{y}_q$. The model we are proposing here has more parameters to be estimated but we have not experienced any robustness or overfitting problems, see section VI. Finally, the update of π_k could be considered the spike and slab counterpart of our γ_{kq} parameter update. π_k is shared by all $\mathbf{y}_q$ and the same is true for γ_s. This is not the case for γ_{kq} in our model.

B. Suboptimal greedy version

Since the Bayesian K-SVD algorithm we introduced in the previous sections takes into account the uncertainties of the coefficients to improve the estimation, it is computationally expensive. As an example, using non-optimized code on a server equipped with Intel Xeon® CPU E5-4640 @ 2.40 GHz processor, the learning and reconstruction phases for a 64×255025 $\mathbf{Y}$ matrix using $Q = 256$ atoms require 3 hours and 30 minutes, respectively. During training and reconstruction, the bottleneck is in the computation of the sparse representation in which atoms are added, deleted or reestimated.

To improve the overall speed of the Bayesian K-SVD algorithm, and in particular, that of the sparse representation computation, we introduce a faster version. Inspired by the approach of greedy algorithms like OMP, we propose to compute the sparse representation in an additive suboptimal fashion. This faster version only adds atoms instead of reestimating or deleting elements in the support of the sparse signals. That is, when the likelihood is maximized only by removing or reestimating a new atom, the sparse representation calculation stops. This approach allows for the whole BKSVD algorithm to perform fewer operations and hence leads to faster iterations.

To validate the proposed approach, we ran the following synthetic experiment. We generated a $\mathbf{D}_{64\times150}$ dictionary and a sparse matrix $\mathbf{X}_{150\times1500}$ with different numbers of non-zero components per column, see Table I, and calculated $\mathbf{Y} = \mathbf{DX}$ with no noise.

We compared the BKSVD algorithm and its faster version by examining the percentage of columns in $\mathbf{X}$ for which each method correctly selects at least 80% of the atoms, as shown in Table I. As can be seen in it, the performance of the two algoritms is comparable for smaller values of s.

We show next in Figure 1 the required computation times using different dictionary sizes K with $K = iP$, $i = 2, 3, 4$, $P = 64$ and the values of Q corresponding to the total number of overlapping patches in $128 \times 128, 256 \times 256$ and 512×512 images assuming full overlap. As can be seen from it, the computational savings are significant. All the experiments we present in Sec. VI are performed using this faster and greedy method.

Finally, we would like to note here that the term *fast* is used in the paper title to refer to the greedy algorithms proposed in sections V.A and V.B to compute the Bayesian inference and not to a property of our method in comparison with other

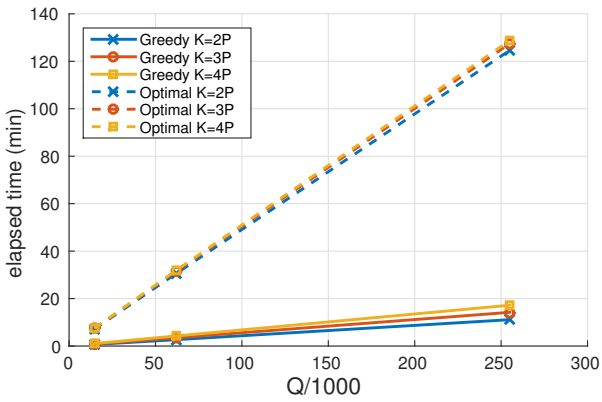


Fig. 1: Time required to compute the sparse representation of synthetic data (as in the experiment in Table I) for the proposed methods.

TABLE I: Performance of the BKSVD algorithm and its faster greedy version for different percentages of non-zero components s

s (%)	Optimal sparse coding	Greedy sparse coding
3	100.00	100.00
6	100.00	99.87
10	99.67	98.50
13	92.00	70.00

learning dictionary techniques. In the experimental section we have reported in an experiment the computational time required by each method.

C. Influence of dictionary size K

The size of the dictionary influences the execution time of the proposed algorithm as well as the accuracy of the representations. In Fig. 2 we can see the influence of K using three different metrics: training time, MSE, and number of atoms used. Experiments were carried out on the 256×256 "Lena" image. We trained different dictionaries of size $P \times kP$, $k = 1, 1.5, \ldots, 8$, with $P = 64$. It is worth mentioning that when $k = 1$ the dictionary becomes a complete one. For each experiment we computed the training time, the reconstruction error $\propto ||\mathbf{Y} - \mathbf{DX}||_{\mathrm{F}}/PQ$, and the mean number of used atoms. We can appreciate a linear trend in the training time figure, which implies that increasing the dictionary size does not have a drastic impact in computation time. On the other hand, as expected, the larger the dictionary, the lower the achieved representation error while maintaining sparsity. Finally, the figure at the bottom shows that by increasing the dictionary size we obtain sparser representations. However, from $k = 2.5$ to $k = 4$ we can see a small plateau where increasing the dictionary size does not have an effect on sparsity. For values of k higher than 4.5 the sparsity keeps increasing but at the expense of a higher computational time due to the unnecessarily large dictionaries.

Since we are interested in finding a good trade-off between error, computational complexity and sparseness of the solution, we should seek a k value for which the algorithm performs

well in a reasonable amount of time. Taking into account the aforementioned behaviour of the considered metrics, we chose $k = 4$ as a good trade-off value. Thus, since the experiments carried out in Sec. VI make use of 8×8 image patches, the resulting dictionary are of size 64×256.

Lastly, it is also worth noting that the level of sparsity we show in this experiment is lower than the one depicted in the denoising experiments. The reason lies in the value of the noise variance: higher variance favors sparser representations by increasing the degrees of freedom of the solution. On the other hand, a very small variance tends to reduce sparsity.

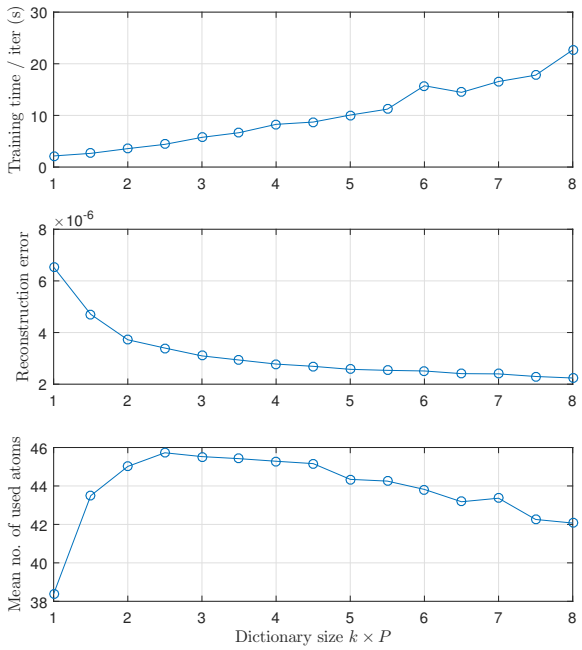


Fig. 2: Influence of K on training time, reconstruction error and sparsity. For these experiments $P = 64$.

VI. Experimental results

In this section we show the results of the experiments on denoising and inpainting we carried out to demonstrate the performance of the proposed BKSVD algorithm on real data. We use standard image processing tasks as a proxy to evaluate the quality of the estimated dictionaries. This is the reason why we compare all algorithms under the same settings by performing only basic dictionary learning operations and no further processing. We assume that all considered applications would benefit from an increased complexity of the method by introducing specific task related operations. As an example, in [59] the authors add a new inner step within the KSVD algorithm which improves the performance for the specific inpainting case.

Experiments were performed on four typical grayscale images, namely *Barbara*, *Boat*, *Lena* and *Peppers*. For both denoising and inpainting a dictionary of 256 atoms was learned. The dictionary was initialized with an overcomplete DCT dictionary. To have an unbiased dictionary, the mean is removed from each patch before running the BKSVD algorithm and then added back to the processed patches. Images are divided into 8×8 overlapping patches vectorized in columns and stacked into a matrix. We use maximum overlap for better performance, although it slows down the representation task. Recovered overlapping patches are then averaged according to their pixel contribution to the image. We used $a^\beta = b^\beta = 1$ for the Gamma hyperprior distribution.

A. Denoising

To assess the performance of the proposed fast BKSVD algorithm, we compare it with K-SVD, BPFA [30] and SnS [35] methods. Differently from K-SVD, which requires knowledge of the exact noise variance, information rarely available in real problems, the proposed Bayesian approach as well as the BPFA and SnS methods are able to infer this quantity directly from the corrupted data. To perform a fair comparison, K-SVD is run with both the noise variance estimated by our method and the true added noise.

We learned the dictionaries for the techniques we consider in this experiment using the noisy patches of the image itself (of size 256×256 pixels). We corrupted the images with additive white Gaussian noise (AWGN) with standard deviation $\sqrt{1/\beta} \in \{5, 10, 15, 20, 25, 50\}$.

We show in Table V a comparison of the techniques. As we have already mentioned, we compared the proposed BKSVD algorithm with the K-SVD algortihm utilizing the true noise standard deviation and the one estimated by our algorithm. Notice that, since our noise estimate is very close to the true one, these two experiments resulted in similar results.

TABLE II: Estimated σ using the "Lena" image.

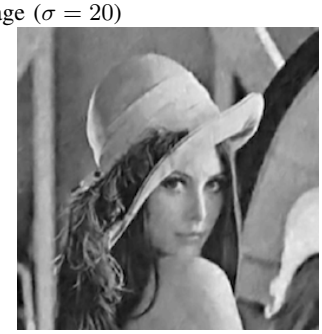
σ	5	10	15	20	25	50
$\hat{\sigma}$	5.54	10.39	15.01	19.55	24.46	46.80

The proposed method performs equally or better than K-SVD in 20 out of 24 experiments and also is capable of estimating the noise variance. Notice also that unlike our method, K-SVD is very sensitive to noise variance mismatch. This mismatch can decrease its PSNR performance by a few dBs [9]. On the other hand, our technique performs a completely automatic noise variance estimation and is more robust to high noise levels because it takes into account the uncertainty of the estimates. We show in Table V the average percentage of non-zero components in the estimated $\mathbf{X}$. As can be seen, while PSNR and SSIM values are similar for both techniques, BKSVD always obtains sparser solutions which indicates that the learned dictionary with our method contains atoms which can better represent the signal. The Spike and Slab (SnS) method in [35] directly compares to our techinque since it also has the ability of automatically estimate the noise variance during the process. In Table V we can see that this method is able to reach competitive results in the denoising task. However, our method outperforms SnS in terms of PSNR and SSIM. By comparing the proposed technique with the one in [30] we can see that the two techniques, which both can automatically estimate the parameters of the model, perform

(a) Noisy image ($\sigma = 20$)

(b) BKSVD denoising

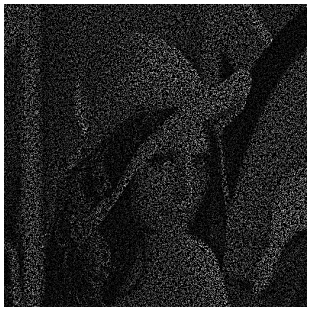
(c) K-SVD denoising



(d) BPFA denoising

(e) SnS denoising

Fig. 3: Comparison of the denoising performance of BKSVD, K-SVD, BPFA [30] and Spike and Slab (SnS) model [35] algorithms.



(a) $r = 75\%$ missing pixels

(b) BKSVD inpainting

(c) K-SVD inpainting

Fig. 4: Comparison of the inpainting performance of BKSVD and K-SVD algorithms. Please note that in this comparison the dictionary is learn from a set of clean images.

almost equally well. The difference in PSNR and SSIM values between the two techniques is, in most cases, extremely small. Thus, in order to further assess in detail the differences in the quality of the estimated dictionaries, we rely on the *mutual coherence* defined as

$$\mu\{\mathbf{D}\} = \max_{i \neq j} \frac{|\mathbf{d}_i^\intercal \mathbf{d}_j|}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|}.$$

The mutual coherence provides a measure of the worst similarity between any two columns in the dictionary; in fact, two columns with high correlation may confuse algorithms seeking a sparse representation [61]. Even though the design of incoherent dictionary is out of the scope of this work, in Table III we show the mutual coherence of the dictionaries learned with the proposed technique and the one in [30] to further analyze the differences among the two techniques. It can be seen that our technique is able to consistently reach lower mutual coherence at different dictionary sizes. We further investigate this aspect by characterizing the quality of the dictionary itself. In order to perform this task, given a set of non-noisy training patches we learn the dictionaries made of 256 atoms using both the proposed method and the BPFA technique [30]. Then, we use these dictionaries to denoise a specific image. In order to perform a fair comparison we use the OMP algorithm for the sparse representation step with a fixed sparsity level $s = 3$. As can be seen in Table IV, the dictionary learned with the proposed algorithm not

(a) BKSVD inpainting (b) K-SVD inpainting (c) BPFA inpainting (d) SnS inpainting

Fig. 5: Comparison of the inpainting performances at $r = 75\%$ of BKSVD, K-SVD [60], BPFA [30] and Spike and Slab (SnS) model [35] algorithms. Please note that in this comparison the dictionary is learnt from the corrupted images themselves.

only has lower mutual coherence but also shows its ability to better sparsely represent a signal by reaching higher PSNR at different noise standard deviations σ.

TABLE III: Coherence comparison between the proposed technique and BPFA [30]. The dictionaries were learned from $5e3$ patches of size 8×8 extracted from a set of natural images. Different dictionary sizes K are considered.

K	64	96	128	160	192	224	256
Proposed	**0.86**	**0.87**	**0.91**	**0.94**	**0.94**	**0.94**	**0.95**
BPFA [30]	0.96	0.97	0.97	0.97	0.98	0.98	0.98

TABLE IV: Denoising performed on noisy *Lena* using OMP by employing the dictionaries learned from a set of $94e3$ patches of size 8×8 extracted from non-noisy natural images. We specifically compare our dictionary with the one learned using the techinque in [30].

σ	5	10	15	20	25	50
Proposed	**33.03**	**32.20**	**31.15**	**29.83**	**28.57**	**23.88**
BPFA [30]	28.48	28.29	28.00	27.52	26.93	23.70

An example of denoising is shown in Figure 3 where we can see the denoised "Lena" provided by the different methods we are considering in this section. As shown in this figure, the proposed technique preserves edges and high spatial frequencies better than K-SVD, which produces a flatter and more blurry image. Moreover, as we already pointed out previously, our method does not require any prior information on the noise corrupting the images since its variance is automatically estimated. When considering other methods, it can also be seen that our method is able to generate less visible high-frequency artifacts than BPFA and SnS models.

Table II shows both the true synthetic noise standard deviation and the corresponding estimation by our method for the denoising experiment using the "Lena" image.

B. Inpainting

Sparse coding is also capable of dealing with missing information. The problem stated in (1) needs to be adapted to handle this lack of information at the reconstruction phase, that is, after the dictionary has been learned.

For this experiment, a dictionary of 256 atoms was learned from a database of 23 images. From every image, we selected the 4096 patches with the highest variances. Following the approach in [7], these images did not contain missing values. During testing, and for images not in the training set, 25%, 50% and 75% of the pixels in those images were removed (set to zero) from every non-overlapping patch of each 512×512 test image. No noise was added. Regarding the K-SVD parameters, a very small σ was used since the image has no noise.

During testing, the process was adapted to deal with the missing information. Let n_q denote the position of the pixels in a patch q where the information is available. We create the set of truncated vectors $\tilde{\mathbf{y}}_q = \mathbf{y}_q(n_q)$ which contain the entries of $\mathbf{y}_q$ restricted to the indices in n_q, and consider the set of truncated dictionaries for these signals $\tilde{\mathbf{D}}^{(q)} = [\mathbf{d}_1(n_q), \cdots, \mathbf{d}_K(n_q)]$. We then estimate $\mathbf{x}_q$ from the observation model

$$\tilde{\mathbf{y}}_q = \tilde{\mathbf{D}}^{(q)}\mathbf{x}_q, \quad q = 1, \ldots, Q. \tag{73}$$

Finally, the image is recovered from the estimated representations $\hat{x}_q$ and the full dictionary, $\hat{\mathbf{Y}} = \mathbf{D}\hat{\mathbf{X}}$. This process is depicted in Figure 6.

As we can see in table VI, the results obtained by the proposed method outperform those obtained by K-SVD, suggesting an improved capability of representation by the learned BKSVD dictionary. Notice that for high percentages r of missing pixels and due to the scarcity of data both methods perform similarly, although the proposed one still performs slightly better. We show a graphical example in figure 4 for the highest ratio of missing pixels ($r = 75\%$). There is a noticeable improvement in the visual quality of the image recovered by our method in contrast to the too smooth K-SVD reconstruction.

BKSVD can be adapted to learn the dictionary from images with missing pixels. In this scenario, the per block observation model becomes

$$\mathbf{M}_q\mathbf{y}_q = \mathbf{M}_q\mathbf{D}\mathbf{x}_q + \mathbf{M}\epsilon_q$$

where $\mathbf{M}_q$ is a diagonal matrix with a value of 1 if the corresponding position of $\mathbf{y}_q$ is observed and 0 otherwise,

TABLE V: Comparison of the proposed BKSVD algorithm with K-SVD [7], batch and online versions of the BPFA algorithm [30], and Spike and Slab model in [35]. K-SVD was tested with estimated and true σ. We show PSNR (dB), SSIM and average sparsity (non-zero coefficients over the number of coefficients) denoted as *nz*. For the SnS model we do not provide the *nz* since the learned dictionary is not overcomplete.

σ	Barbara			Boats			Lena			Peppers			
	PSNR	SSIM	nz	PSNR	SSIM	nz	PSNR	SSIM	nz	PSNR	SSIM	nz	
5	38.02	**0.97**	0.05	**36.71**	**0.96**	**0.03**	38.90	**0.97**	0.04	**38.98**	**0.97**	0.04	*BKSVD*
	38.00	**0.97**	0.08	36.70	**0.96**	0.08	38.92	**0.97**	0.08	38.91	**0.97**	0.07	*K-SVD $\hat{\sigma}$*
	38.10	**0.97**	0.10	36.40	**0.96**	0.12	**38.93**	**0.97**	0.08	38.89	**0.97**	0.08	*K-SVD σ*
	35.47	0.94	**0.03**	36.11	**0.96**	**0.03**	37.77	**0.97**	**0.03**	38.02	**0.97**	**0.03**	*BPFA [30]*
	35.99	0.95	-	36.44	**0.96**	-	38.21	**0.97**	-	38.32	0.97	-	*SnS [35]*
10	**33.97**	**0.93**	**0.02**	32.80	0.91	**0.02**	**34.96**	**0.94**	**0.02**	**35.27**	**0.94**	**0.01**	*BKSVD*
	33.95	**0.93**	0.04	32.70	0.90	0.04	34.94	**0.94**	0.04	35.08	**0.94**	0.03	*K-SVD $\hat{\sigma}$*
	33.97	**0.93**	0.06	33.00	**0.92**	0.04	34.95	**0.94**	0.04	35.09	**0.94**	0.02	*K-SVD σ*
	32.97	0.91	**0.02**	33.00	**0.92**	**0.02**	34.75	**0.94**	**0.02**	35.15	**0.94**	0.02	*BPFA [30]*
	32.98	0.92	-	**32.88**	**0.92**	-	34.59	**0.94**	-	34.81	**0.94**	-	*SnS [35]*
15	**31.85**	**0.90**	**0.01**	31.01	0.87	**0.01**	**32.78**	**0.91**	**0.01**	**33.12**	**0.92**	**0.01**	*BKSVD*
	31.82	**0.90**	0.02	30.90	0.87	0.03	32.73	**0.91**	0.03	33.00	**0.92**	0.03	*K-SVD $\hat{\sigma}$*
	31.84	**0.90**	0.02	31.00	0.87	0.02	32.70	**0.91**	0.03	33.04	**0.92**	0.02	*K-SVD σ*
	31.43	0.88	**0.01**	**31.06**	**0.88**	0.02	32.71	**0.91**	**0.01**	33.10	**0.92**	**0.01**	*BPFA [30]*
	30.87	0.87	-	30.73	0.87	-	32.33	0.90	-	32.63	0.91	-	*SnS [35]*
20	**30.30**	**0.87**	**0.01**	29.44	0.83	**0.01**	**31.32**	0.88	**0.01**	31.36	0.89	**0.01**	*BKSVD*
	30.24	**0.87**	0.02	29.42	0.83	0.02	31.27	0.88	0.02	31.35	0.89	0.02	*K-SVD $\hat{\sigma}$*
	30.28	**0.87**	0.02	29.44	0.83	0.02	31.30	0.88	0.02	31.36	0.89	0.02	*K-SVD σ*
	30.01	0.85	**0.01**	**29.56**	**0.84**	**0.01**	31.19	0.88	**0.01**	**31.56**	**0.90**	**0.01**	*BPFA [30]*
	29.52	0.84	-	29.30	0.83	-	30.76	0.87	-	31.01	0.89	-	*SnS [35]*
25	**29.10**	**0.84**	**0.01**	28.38	0.80	**0.01**	29.99	**0.86**	**0.01**	30.20	**0.88**	**0.01**	*BKSVD*
	29.10	0.83	0.02	28.38	0.80	0.02	29.87	**0.86**	0.02	30.12	0.87	0.02	*K-SVD $\hat{\sigma}$*
	29.05	0.83	**0.01**	28.36	0.80	**0.01**	30.00	**0.86**	**0.01**	30.15	0.87	0.01	*K-SVD σ*
	29.01	0.83	**0.01**	**28.63**	**0.81**	**0.01**	**30.12**	**0.86**	**0.01**	**30.27**	**0.88**	**0.01**	*BPFA [30]*
	28.49	0.81	-	28.13	0.79	-	29.56	0.84	-	29.82	0.86	-	*SnS [35]*
50	25.60	**0.72**	**0.01**	25.10	0.67	**0.01**	26.31	0.73	**0.01**	26.41	0.77	**0.01**	*BKSVD*
	25.51	0.71	**0.01**	25.01	0.66	**0.01**	26.16	0.72	**0.01**	26.19	0.76	**0.01**	*K-SVD $\hat{\sigma}$*
	25.43	0.71	**0.01**	24.92	0.67	**0.01**	26.23	0.72	**0.01**	26.32	0.77	**0.01**	*K-SVD σ*
	25.78	**0.72**	**0.01**	**25.26**	**0.68**	**0.01**	**26.43**	**0.76**	**0.01**	**26.71**	**0.80**	**0.01**	*BPFA [30]*
	25.33	0.70	-	24.90	0.66	-	26.13	0.73	-	26.16	0.77	-	*SnS [35]*

TABLE VI: Inpainting results when the dictionary is learnt from a set of natural images. Comparison of the proposed BKSVD algorithm with K-SVD for different ratios (r) of missing pixels. PSNR and SSIM values are given.

$r(\%)$	Barbara		Boat		Lena		Peppers		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
25	**39.18**	**0.99**	**38.54**	**0.97**	**41.54**	**0.98**	**39.49**	**0.97**	BKSVD
	38.80	**0.99**	36.79	0.96	40.58	0.97	37.17	0.97	K-SVD [7]
50	**33.15**	**0.96**	**32.65**	**0.92**	**36.14**	**0.95**	**34.73**	**0.93**	BKSVD
	32.59	0.95	32.25	0.90	36.02	0.94	33.82	0.89	K-SVD [7]
75	**27.36**	**0.86**	**27.39**	**0.80**	**30.37**	**0.88**	**29.29**	**0.85**	BKSVD
	25.03	0.82	26.93	0.76	29.23	0.85	28.70	0.83	K-SVD [7]

and $\epsilon_q \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I})$. All the prior models remain the same. Inference and fast inference can be carried out following the steps in sections IV and V-A, respectively.

Table VII shows a comparison of BKSVD with the methods in [60], [30] and [35]. These methods learn the dictionary from the corrupted image. The same missing pixel patterns as in the previous experiments were used for testing this second approach. The BKSVD inpainting algorithm performs similarly to [60], [30] and [35] in terms of PSNR and SSIM. However, its visual quality is better, with fewer noticeable artifacts on the edges. This can be observed in Fig. 5, BPFA and SnS images, have a higher PSNR values, but visually are worse. Notice also the visual artifacts in the K-SVD reconstruction.

Finally, we would like to mention that although the obtained PSNR values for both experiments (learning from clean or corrupted images) are similar, the visual quality of the reconstructed images is much better in the first case, where the use of additional clean data improves the construction of the dictionary.

C. Runtime comparison

We conclude the experimental section with a final comparison of the dictionary learning algorithms in terms of execution time. The experiments were performed on a server equipped with AMD Opteron™CPU @ 2.30 GHz processors. Table VIII shows runtime for denoising performed on the Lena image with $\sigma = 20$ for different image and dictionary sizes.

K-SVD is the fastest, but it does not include an estimation of the number of non-zeros in each sparse representation. Among the Bayesian methods the proposed method ranks second after BFPA but notice, as pointed out in section V-A,

TABLE VII: Inpainting results when the dictionary is learnt from the corrupted images themselves. Comparison of the proposed BKSVD algorithm with SnS and BPFA for different ratios (r) of missing pixels. PSNR and SSIM values are given.

$r(\%)$	Barbara		Boat		Lena		Peppers		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
25	38.53	**0.98**	36.28	**0.97**	38.85	0.97	39.04	0.97	BKSVD
	38.85	**0.98**	**36.83**	0.96	**39.14**	0.97	**40.14**	0.97	K-SVD [60]
	34.77	0.94	35.19	0.96	37.87	0.97	38.04	0.97	BPFA [30]
	36.34	0.96	35.24	0.95	38.52	**0.98**	39.03	**0.98**	SnS [35]
50	**33.67**	**0.95**	30.97	0.92	33.73	0.95	33.92	**0.96**	BKSVD
	32.58	0.91	30.73	0.88	33.09	0.90	33.48	0.91	K-SVD [60]
	33.33	0.94	**32.24**	**0.93**	**34.43**	**0.96**	35.10	**0.96**	BPFA [30]
	32.65	0.93	31.21	0.91	33.89	0.95	34.69	**0.96**	SnS [35]
75	27.99	**0.86**	25.88	0.79	28.18	0.87	27.84	0.88	BKSVD
	23.48	0.70	22.84	0.62	24.13	0.69	23.16	0.70	K-SVD [60]
	28.24	0.85	**26.91**	**0.83**	**29.46**	**0.91**	**29.36**	**0.92**	BPFA [30]
	27.57	0.82	26.45	0.79	28.93	0.89	28.79	0.90	SnS [35]

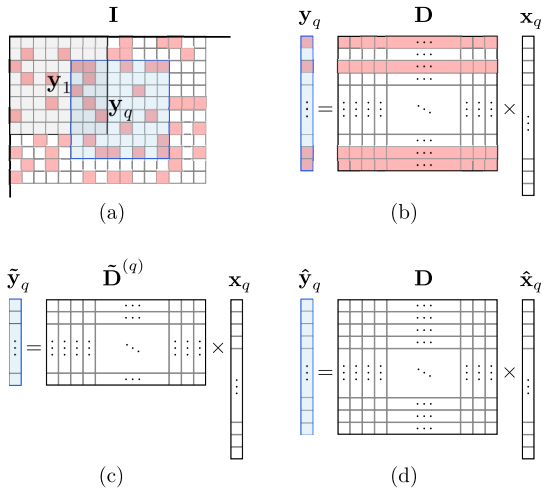


Fig. 6: Inpainting process: (a) two patches from image $\mathbf{I}$, $\mathbf{y}_1$ and $\mathbf{y}_q$ (missing pixels in red), (b) vectorization of patch $\mathbf{y}_q$; rows from $\mathbf{D}$ corresponding to the missing pixels in $\mathbf{y}_q$ are also highlighted in red, (c) highlighted entries are discarded from the problem formulation, (d) recovery using the full dictionary $\mathbf{D}$.

TABLE VIII: Runtime in seconds for dictionary learning algorithms for different image and dictionary (K) sizes.

Image size	K	BKSVD	BPFA	K-SVD	SnS
256x256	256	2.843	1.054	165	9.960
	320	3.603	1.468	171	15.355
128x128	256	774	281	62	2.451
	320	894	407	71	3.893

that the proposed method estimates a parameter λ_{kq} for every coefficient in the representation matrix, whereas BFPA only estimates one probability vector (K parameters) shared by all $\mathbf{x}_q$. It is possible to estimate c such probability vectors in BFPA, but this was not used in the experiments.

VII. Conclusion

In this paper, we presented a novel Bayesian approach for the ℓ_1 sparse dictionary learning problem based on K-SVD. The prior we utilize on the sparse signals enforces sparsity while allowing for a tractable Bayesian inference. The use of Bayesian modeling and inference allows us to take into account the uncertainty of the estimates in the inference process. Very importantly, the proposed technique estimates all parameters without the need of any additional information, which makes it fully automatic.

The proposed algorithm has been tested on denoising and inpainting tasks. For the inpainting problem, dictionaries were learned using two different approaches; first, utilizing a set of clean images, and secondly, from the corrupted images themselves. All evaluated techniques yield good results, but K-SVD has a major drawback: the need of an accurate estimate of the noise deviation. Bayesian methods solve this deficiency, but at the expense of higher complexity and computational time. From the performed experiments, BPFA and the proposed method are the best performing ones; notice also that our method produces images with reduced artifacts. In order to further characterize the improved quality of our dictionary estimation, we evaluated it in terms of lower mutual coherence and better image recovery under common sparse presentation algorithms.

Lastly, experiments to analyze the importance of the dictionary size have been carried out. These experiments can be used to tune the size of the dictionary. We have not addressed here the estimation of the size of the dictionary using Bayesian inference. However, we are currently investigating how to adapt to our model the approach to initially estimate the dictionary size presented in [30].

References

[1] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[2] A. Eftekhari and M. B. Wakin, *New Analysis of Manifold Embeddings and Signal Recovery from Compressive Measurements*. Elsevier Inc., 2013, vol. 39, no. 1.

[3] B. R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for Sparse Representation Modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[4] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[5] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[6] ——, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[7] 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, p. 4311, 2006.

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on MachineLearning*, 2009, pp. 689–696.

[9] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-parametric Bayesian dictionary learning for sparse image representations," *IEEE Transactions on Image Processing*, 2009.

[10] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A non-convex relaxation approach to sparse dictionary learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1809–1816.

[11] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[12] M. Elad, M. A. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[13] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[14] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[15] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.

[16] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3501–3508.

[17] T. H. Vu, H. S. Mousavi, V. Monga, G. Rao, and U. K. A. Rao, "Histopathological image classification using discriminative feature-oriented dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 3, pp. 738–751, March 2016.

[18] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[19] K. Engan, K. Skretting, and J. H. Husø y, "Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digital Signal Processing: A Review Journal*, vol. 17, no. 1, pp. 32–49, 2007.

[20] Q. Liu, S. Wang, K. Yang, J. Luo, Y. Zhu, and D. Liang, "2013 Qiegen LiuHighly Undersampled Magnetic Resonance Image Reconstruction Using Two-Level Bregman Method With Dictionary Updating," vol. 32, no. 7, pp. 1290–1301, 2013.

[21] Q. Liu, D. Liang, Y. Song, J. Luo, Y. Zhu, and W. Li, "Augmented Lagrangian-Based Sparse Representation Method with Dictionary Updating for Image Deblurring," vol. 6, no. 3, pp. 1689–1718, 2013.

[22] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.

[23] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Z.-f. Han, and J. Chambers, "Analysis SimCO Algorithms for Sparse Analysis Model Based Dictionary Learning," *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 417–431, 2016.

[24] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations." *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[25] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.

[26] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation." *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[27] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[28] D. M. Bradley and J. A. Bagnell, "Differential Sparse Coding," *Neural Information Processing Systems*, 2008.

[29] M. Girolami, "A variational method for learning sparse and overcomplete representations." *Neural computation*, vol. 13, no. 11, pp. 2517–2532, 2001.

[30] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[31] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 777–784.

[32] D. Knowles and Z. Ghahramani, "Infinite sparse factor analysis and infinite independent components analysis," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 381–388.

[33] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X.-p. Zhang, "Bayesian Nonparametric Dictionary Learning for Compressed Sensing MRI," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5007–5019, 2013.

[34] H. Ishwaran and J. S. Rao, "Spike and slab variable selection: frequentist and bayesian strategies," *Annals of Statistics*, pp. 730–773, 2005.

[35] M. Lázaro-Gredilla and M. K. Titsias, "Spike and slab variational inference for multi-task and multiple kernel learning," in *Advances in neural information processing systems*, 2011, pp. 2339–2347.

[36] R. Yoshida and M. West, "Bayesian learning in sparse graphical factor models via variational mean-field annealing," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1771–1798, 2010.

[37] A.-S. Sheikh, J. A. Shelton, and J. Lücke, "A truncated em approach for spike-and-slab sparse coding." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2653–2687, 2014.

[38] L. Li, J. Silva, M. Zhou, and L. Carin, "Online Bayesian dictionary learning for large datasets," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 2157–2160.

[39] M. R. Andersen, O. Winther, and L. K. Hansen, "Bayesian inference for structured spike and slab priors," in *Advances in Neural Information Processing Systems*, 2014, pp. 1745–1753.

[40] M. R. Andersen, A. Vehtari, O. Winther, and L. K. Hansen, "Bayesian inference for spatio-temporal spike and slab priors," *ArXiv e-prints*, Sep. 2015.

[41] H. S. Mousavi, V. Monga, and T. D. Tran, "Iterative convex refinement for sparse recovery," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1903–1907, Nov 2015.

[42] T.-J. Yen, "A majorization-minimization approach to variable selection using spike and slab priors," *The Annals of Statistics*, pp. 1748–1775, 2011.

[43] Y. Zhang, R. Henao, C. Li, and L. Carin, "Bayesian dictionary learning with gaussian processes and sigmoid belief networks," *unpublished*, 2016.

[44] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 399–406.

[45] A. Szlam, K. Gregor, and Y. LeCun, *Fast Approximations to Structured Sparse Coding and Applications to Object Classification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 200–213.

[46] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 883–891, 2014.

[47] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[48] R. Gribonval, R. Jenatton, and F. Bach, "Sparse and Spurious: Dictionary Learning With Noise and Outliers," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6298–6319, 2015.

[49] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1356–1369, 2016.

[50] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Verlag, 2010.

[51] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[52] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 53–63, Jan 2010.

[53] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 341–355.

[54] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006.

[55] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003, pp. 3–6.
[56] Z. Chen, R. Molina, and A. K. Katsaggelos, "Automated recovery of compressedly observed sparse signals from smooth background," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 1012–1016, Aug 2014.
[57] S. D. Babacan, L. Mancera, R. Molina, and A. K. Katsaggelos, "Bayesian Compressive Sensing Using Non-Convex Priors," *European Signal Processing Conference 2009 EUSIPCO09*, 2009.
[58] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
[59] M. S. Koh and E. Rodriguez-Marek, "Turbo inpainting: Iterative k-svd with a new dictionary," in *Multimedia Signal Processing, 2009. MMSP '09. IEEE International Workshop on*, Oct 2009, pp. 1–6.
[60] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer Publishing Company, Incorporated, 2010.
[61] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2055–2065, April 2013.

Juan G. Serra received the degree in Telecommunications Engineering from the Universitat Politècnica de València in 2014 and the M.S. degree in Data Science and Computer Engineering from the University of Granada in 2016. He started the Ph.D. degree in 2015 at the University of Granada under the supervision of Prof. Molina, being a member of the Visual Information Processing Group in the Department of Computer Science and Artificial Intelligence. His research interests focus on the use of Bayesian modeling and inference to solve different inverse problems related to dictionary learning, image restoration and machine learning. During his research, he has addressed several problems, such as blind image deconvolution, image denoising and inpainting and multispectral image classification.

Matteo Testa received the B. Sc. degree and the M. Sc. degree in telecommunications engineering from the Politecnico di Torino, Turin, Italy, in 2011 and 2012, respectively, and the Ph.D. degree in electronic and communications engineering from the Electronics Department, Politecnico di Torino, in 2016, under the supervision of Prof. E. Magli. He currently holds a post-doctoral position with the IPL lab, Politecnico di Torino, led by Prof. E. Magli in collaboration with SONY EuTEC. His research is focused on compressed sensing with a particular interest on imaging systems, forensic applications and bayesian inference.

Rafael Molina was born in 1957. He received the degree in mathematics (statistics) and the Ph.D. degree in optimal design in linear models from the University of Granada, Granada, Spain, in 1979 and 1983, respectively. He became Professor of Computer Science and Artificial Intelligence at the University of Granada, Granada, Spain, in 2000. He is the former Dean of the Computer Engineering School at the University of Granada (19922002) and Head of the Computer Science and Artificial Intelligence department of the University of Granada (20052007). His research interest focuses mainly on using Bayesian modeling and inference in problems like image restoration (applications to astronomy and medicine), superresolution of images and video, blind deconvolution, computational photography, source recovery in medicine, compressive sensing, low-rank matrix decomposition, active learning, fusion, and classification.

Prof. Molina serves as an Associate Editor of Applied Signal Processing (20052007); the IEEE Transactions on Image Processing (2010present); and Progress in Artificial Intelligence (2011present); and an Area Editor of Digital Signal Processing (2011present). He is the recipient of an IEEE International Conference on Image Processing Paper Award (2007) and an ISPA Best Paper Award (2009). He is a coauthor of a paper awarded the runner-up prize at Reception for early-stage researchers at the House of Commons.

Aggelos K. Katsaggelos (F98) received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in Electrical Engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. In 1985, he joined the Department of Electrical Engineering and Computer Science at Northwestern University, where he is currently a Professor, holder of the Joseph Cummings Chair. He is a member of the Academic Staff, NorthShore University Health System, an affiliated faculty at the Department of Linguistics and he has an appointment with the Argonne National Laboratory. He was previously the holder of the Ameritech Chair of Information Technology and the AT&T Chair and the co-Founder and Director of the Motorola Center for Seamless Communications. He has published extensively in the areas of multimedia processing and communications (over 250 journal papers, 500 conference papers and 40 book chapters) and he is the holder of 26 international patents. He is the co-author of Rate-Distortion Based Video Compression (Kluwer, 1997), Super-Resolution for Images and Video (Claypool, 2007), Joint Source-Channel Video Transmission (Claypool, 2007), Machine Learning Refined (Cambridge University Press, 2016) and The Essentials of Sparse Modeling and Optimization (Springer, 2017, forthcoming). He has supervised 56 PhD dissertations so far.

Among his many professional activities Prof. Katsaggelos was Editor-in-Chief of the IEEE Signal Processing Magazine (1997-2002), a BOG Member of the IEEE Signal Processing Society (1999-2001), a member of the Publication Board of the IEEE Proceedings (2003-2007), and he is currently a member of the Awards Board of the Signal Processing Society. He is a Fellow of the IEEE (1998) and SPIE (2009) and the recipient of the IEEE Third Millennium Medal (2000), the IEEE Signal Processing Society Meritorious Service Award (2001), the IEEE Signal Processing Society Technical Achievement Award (2010), an IEEE Signal Processing Society Best Paper Award (2001), an IEEE ICME Paper Award (2006), an IEEE ICIP Paper Award (2007), an ISPA Paper Award (2009), and a EUSIPCO paper award (2013). He was a Distinguished Lecturer of the IEEE Signal Processing Society (2007-2008).