# On the Convergence of a Bayesian Algorithm for Joint Dictionary Learning and Sparse Recovery

Geethu Joseph and Chandra R. Murthy *Senior Member, IEEE*

***Abstract*—Dictionary learning (DL) is a well-researched problem, where the goal is to learn a dictionary from a finite set of noisy training signals, such that the training data admits a sparse representation over the dictionary. While several solutions are available in the literature, relatively little is known about their convergence and optimality properties. In this paper, we make progress on this problem by analyzing a Bayesian algorithm for DL. Specifically, we cast the DL problem into the sparse Bayesian learning (SBL) framework by imposing a hierarchical Gaussian prior on the sparse vectors. This allows us to simultaneously learn the dictionary as well as the parameters of the prior on the sparse vectors using the expectation-maximization algorithm. The dictionary update step turns out to be a non-convex optimization problem, and we present two solutions, namely, an alternating minimization (AM) procedure and an Armijo line search (ALS) method. We analytically show that the ALS procedure is globally convergent, and establish the stability of the solution by characterizing its limit points. Further, we prove the convergence and stability of the overall DL-SBL algorithm, and show that the minima of the cost function of the overall algorithm are achieved at sparse solutions. As a concrete example, we consider the application of the SBL-based DL algorithm to image denoising, and demonstrate the efficacy of the algorithm relative to existing DL algorithms.**

***Index Terms*—Sparse representation, dictionary learning, non-convex optimization.**

## I. Introduction

In sparse coding, the signal of interest is represented as a linear combination of a relatively small number of columns of a properly chosen over-complete *dictionary*. The dictionary can be of two types: first, non-adaptive or predefined dictionaries like Fourier, Gabor, discrete cosine transform and wavelet [1]; and second, an adaptive or learned dictionary that is specific to the given class of signals. The use of adaptive dictionaries often leads to more compact representations and better performance in many signal processing applications ranging from image denoising [2]–[4], audio processing [5], [6], and classification tasks [7]–[13], to name a few. Therefore, we are interested in the dictionary learning problem, where the objective is to find a dictionary over which a set of training signals admits a sparse representation.

Several dictionary learning algorithms for sparse coding have been proposed in the literature such as method of optimal directions (MOD) [14], K-singular value decomposition (K-SVD) [15], dictionary learning via the majorization method (DL-MM) [16], simultaneous codeword optimization (SimCO) [17], parallel atom-updating dictionary learning (PAU-DL) [18], sequential generalization of K-means (SGK) [19], iterative thresholding and K means (ITKM) [20]. Most of the algorithms involve an iterative procedure, alternately updating the dictionary and the sparse representation, and differ in the cost function used in the dictionary update step. To update the sparse representation, an existing standard sparse signal recovery algorithm is used.

Although the aforementioned algorithms achieve good performance, they require the knowledge of the sparsity level of the system and hand-tuning of various sensitive algorithm parameters. These limitations are handled to some extent by Bayesian algorithms [21]–[24]. Bayesian algorithms come with an added advantage of not requiring the knowledge of the measurement noise variance. However, the posterior distributions proposed in [21], [23], [25]–[27] cannot be derived analytically, and a Gibbs sampler is used for Bayesian inference. The Gibbs sampling based algorithms are computationally demanding as they involve ensemble learning. To overcome this difficulty, [23] also proposes a variational Bayes' based algorithm for dictionary learning by imposing a Gaussian prior on the dictionary elements. The Gaussian prior intuitively models the boundedness of the dictionary elements and helps to obtain closed form expressions for the dictionary update. The closed form expressions results in faster convergence than the Gibbs sampling based Bayesian algorithms. Nonetheless, imposing a Gaussian prior (on a dictionary with no special structure) results in low accuracy and requires a large number of iterations to converge. Therefore, the choice of Gaussian prior still leaves room for improvement. This motivates us to develop an improved Bayesian dictionary learning algorithm which does not require the knowledge of the sparsity level, or fine-tuning of parameters, while at the same time improving on the recovery performance.

Our proposed dictionary learning algorithm is based on the sparse Bayesian learning (SBL) framework [28], [29]. In the context of sparse signal recovery, SBL is known to offer superior performance compared to algorithms based on convex relation and greedy approaches, and does not require one to tune the algorithm parameters. The basic idea of SBL is to incorporate a parameterized prior on the unknown sparse vectors that encourages sparsity. Specifically, a fictitious Gaussian prior is imposed on the sparse vectors, and the so-called hyperparameters of the Gaussian distribution are determined using Type-II maximum likelihood (ML) estimation. Our

G. Joseph is at the Dept. of EECS, Syracuse University, New York, USA 13244. She was at the Dept. of ECE, Indian Institute of Science, Bangalore, India 560012, during the course of this work. Email: geethu@iisc.ac.in. C. R. Murthy is at the Dept. of ECE, Indian Institute of Science, Bangalore 560012, India. Email: cmurthy@iisc.ac.in.

The work of G. Joseph was supported in part by the Intel India PhD fellowship. This work was financially supported in part by the Young Faculty Research Fellowship from the Ministry of Electronics and Information Technology, Govt. of India.

approach is different from other Bayesian dictionary learning algorithms as we impose *no prior on the dictionary elements*. Instead, we estimate the dictionary as a deterministic matrix with unit norm columns. The estimation method uses the expectation-maximization (EM) algorithm to simultaneously learn the parameters of the prior and the sparsifying dictionary. The dictionary update step in the EM algorithm turns out to be a quadratic optimization problem with unit norm constraints, which is a non-convex problem because of the constraint. Since a closed form solution is not available, we propose to employ the alternating minimization (AM) procedure or Armijo line search (ALS) to solve it. Our main contributions are as follows:

- *Algorithm development:* We present a novel algorithm for learning the sparsifying dictionary along with the sparse representations, in Section II.
- *Convergence guarantees for optimization procedures:* We derive convergence guarantees of the dictionary update step using AM and ALS optimization procedures in Section III. We show that the ALS procedure globally converges. We also establish stability of the limit points of the ALS procedure. The results hold irrespective of the sparsity level, the initialization of the algorithm, or the system dimensions.
- *Convergence guarantees for DL-SBL:* We derive convergence guarantees for the entire algorithm, and also discuss about the stability of the limit points in Section IV-A. We show that the DL-SBL cost converges to a single point, while the iterates converge to the set of stationary points.
- *Cost function analysis:* We extend the theoretical guarantees available for the original SBL algorithm [29] to the DL-SBL setting and analyze the SBL cost function in Section IV-B. This analysis shows why the DL-SBL algorithm is likely to converge to the sparsest possible representation of the measurement vectors.
- *Empirical Validation:* In Section V, we empirically corroborate the convergence results in Section III. Further, we illustrate the performance of the algorithms in terms of the relative mean squared error and support recovery rate of the sparse vectors, and Frobenius norm of estimation error and atom recovery rate of the learned dictionary. We compare the proposed scheme with the other popular algorithms when applied to the image denoising problem.

Overall, the proposed algorithm is useful for learning a sparsifying overcomplete dictionary using a given set of training signals. Our algorithm does not require the knowledge of the sparsity level of the system[1] or hand-tuning of the parameters.[2] Finally, the main attraction of our algorithm is the associated theoretical guarantees. Unlike similar existing dictionary learning approaches, we provide rigorous theoretical guarantees for the optimization procedures and the overall algorithm.

**Notation:** Boldface small letters denote vectors and boldface capital letters denote matrices. The $i^{\text{th}}$ entry of a vector $\boldsymbol{a}$ is denoted by $\boldsymbol{a}[i]$, while $\boldsymbol{A}_i$ and $\boldsymbol{A}_{ij}$ represent the $i^{\text{th}}$ column and $(i,j)^{\text{th}}$ entry of a matrix $\boldsymbol{A}$, respectively. The symbol $\|\cdot\|$ denotes the $\ell_2$ norm of a vector or the Frobenius norm of a matrix, and $\|\cdot\|_0$ denotes $\ell_0$ pseudo-norm that counts the number of nonzero entries in a vector. The symbols $(\cdot)^{\mathsf{T}}$, $|\cdot|$, $\mathsf{Tr}\{\cdot\}$ and $(\cdot)^{\dagger}$ denote the transpose, the determinant, the trace and the pseudo-inverse of a matrix, respectively. Also, $\mathsf{Diag}\{\cdot\}$ represents a vector formed using the diagonal entries of a square matrix or a diagonal matrix with entries of the argument vector on the diagonal, depending on the context, and $\mathcal{D}\{\cdot\} = \mathsf{Diag}\{\mathsf{Diag}\{\cdot\}\}$ represents a diagonal matrix with the same diagonal entries as the argument matrix. The pdf of the random variable $X$ is represented as $p(x)$. The expectation with respect to a random variable $X$ is denoted as $\mathbb{E}_X(\cdot)$. The notation $\boldsymbol{I}$, $\boldsymbol{0}$ and $\boldsymbol{1}$ represent the identity matrix, the all zero matrix (or vector), the all ones vector. Also, $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{N}$ denote the set of real numbers, the set of all nonnegative real numbers, and the set of natural numbers, respectively. Throughout the paper, $\boldsymbol{\Gamma} = \mathsf{Diag}\{\boldsymbol{\gamma}\}$, and we use the notations $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma}$ interchangeably, depending on whether we need the matrix or vector version, respectively.

## II. SBL Based Dictionary Learning

We consider a problem setup where we have a set of $K$ training signals $\boldsymbol{y}^K = \{\boldsymbol{y}_k \in \mathbb{R}^m\}_{k=1}^K$ such that $\boldsymbol{y}^K$ admits a sparse representation $\boldsymbol{x}^K = \{\boldsymbol{x}_k \in \mathbb{R}^N\}_{k=1}^K$ over an unknown dictionary $\boldsymbol{A} \in \mathbb{R}^{m\times N}$ and is corrupted by noise, i.e.,

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{1}$$

where the noise term $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. To resolve the ambiguity in amplitude, we assume $\boldsymbol{A}$ has unit norm columns. That is, $\boldsymbol{A} \in \mathbb{O}$, where

$$\mathbb{O} \triangleq \left\{\boldsymbol{A} \in \mathbb{R}^{m\times N} : \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i = 1, i = 1, 2, \ldots, N\right\}. \tag{2}$$

Our goal is to estimate the $K$ sparse vectors and the measurement matrix $\boldsymbol{A}$, using the knowledge of $N$.

Motivated by the SBL framework [28], [29], we impose a fictitious Gaussian prior on the unknown sparse vectors $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{0}, \mathsf{Diag}\{\boldsymbol{\gamma}_k\})$, where $\boldsymbol{\gamma}_k \in \mathbb{R}_+^N$. It is known that the use of Gaussian prior encourages sparsity. On the other hand, we do not assume any structure in $\boldsymbol{A}$ apart from the unit norm columns, and thus, we do not impose any prior on $\boldsymbol{A}$. We note that imposing no prior is equivalent to the use of a uniform prior, i.e., that the columns of $\boldsymbol{A}$ are uniformly distributed on the $m-$dimensional unit sphere. Using this hierarchical model, we first compute the ML estimates $\hat{\boldsymbol{\gamma}}_k$ and $\hat{\boldsymbol{A}}$ of $\boldsymbol{\gamma}_k$ and $\boldsymbol{A}$, respectively. These estimates, in turn, can be used to estimate the sparse vectors as $\hat{\boldsymbol{x}}_k = \mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}_k, \hat{\boldsymbol{\gamma}}_k, \hat{\boldsymbol{A}}\right\}$.

To obtain the ML estimates $\hat{\boldsymbol{\gamma}}_k$ and $\hat{\boldsymbol{A}}$, we need to maximize $p(\boldsymbol{y}^K; \boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda} = \{\boldsymbol{A}, \boldsymbol{\gamma}_k;\ k = 1, 2, \ldots K\} \in \mathbb{O} \times \mathbb{R}_+^{NK}$ is the tuple of unknown parameters.

We now develop an EM procedure to solve the ML estimation problem, equivalently, for minimizing the negative log

[1] We present a version of algorithm that takes the noise variance as an input. A modified version of algorithm which can learn the noise level is provided in [30, Appendix D.11]

[2] The AM-based algorithm does not have any associated parameters, and the recovery performance of the ALS-based algorithm is not sensitive to the choice of its parameters.

likelihood $-\log p(\boldsymbol{y}^K;\boldsymbol{\Lambda})$. Thus, the optimization problem to be solved is $\underset{\boldsymbol{\Lambda}\in\mathbb{O}\times\mathbb{R}_+^{NK}}{\arg\min}\ T(\boldsymbol{\Lambda})$, where the cost function[3] is

$$T(\boldsymbol{\Lambda}) \triangleq \sum_{k=1}^{K} \log\left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| + \boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k. \tag{3}$$

The second term in the cost function, which depends on $\boldsymbol{y}_k$, is equal to $\mathsf{Tr}\left\{\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\boldsymbol{y}_k^{\mathsf{T}}\right\}$. When $\sigma$ is close to zero, this term can be minimized by choosing a column of $\boldsymbol{A}$ to match $\boldsymbol{y}_k$ and driving the corresponding entry of $\boldsymbol{\Gamma}_k$ to infinity. Therefore, the second term of the cost function helps to learn the term $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}$ that matches the measurements. However, it may not be possible to accomplish this for every $\boldsymbol{y}_k$ with a single $\boldsymbol{A}$ matrix. Also, making the entries of $\boldsymbol{\Gamma}_k$ large increases the cost in the first term. Furthermore, there could be multiple $\boldsymbol{\Gamma}_k$ corresponding to the same $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}$, and we are interested in a sparse solution. The true $\boldsymbol{\Gamma}_k$ is thus learned with the help of the first term involving the log det in the above cost function. The log term is minimized when the rank of $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}$ goes to zero. However, if the dimension of the span of $\boldsymbol{y}^K$ is sufficiently large, the second term ensures that $\boldsymbol{A}$ has full row rank. Hence, the log term tries to minimize the rank of $\boldsymbol{\Gamma}_k$. Since $\boldsymbol{\Gamma}_k$ is diagonal, this enforces sparsity in $\boldsymbol{\gamma}_k$, which in turn enforces sparsity in $\boldsymbol{x}_k$. Thus, the two terms of the cost function balance the sparsity and the error in the matching $\boldsymbol{y}_k$ using $\boldsymbol{A}$ and $\boldsymbol{x}_k$.

The EM algorithm treats the unknowns $\boldsymbol{x}^K$ as the hidden data and the observations $\boldsymbol{y}^K$ as known data. It is an iterative procedure with two steps: an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{\Lambda}^{(r)}$ be the estimate of $\boldsymbol{\Lambda}$ at the $r^{\text{th}}$ iteration. The E-step computes the marginal log-likelihood of the observed data $Q^{(r-1)}$, and the M-step computes the parameter tuple $\boldsymbol{\Lambda}$ that maximizes $Q^{(r-1)}$.

$$\textbf{E-step: } Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right) = \mathbb{E}_{\boldsymbol{x}^K|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}}\left\{\log p\left(\boldsymbol{y}^K,\boldsymbol{x}^K;\boldsymbol{\Lambda}\right)\right\}$$

$$\textbf{M-step: } \boldsymbol{\Lambda}^{(r)} = \underset{\boldsymbol{\Lambda}\in\mathbb{O}\times\mathbb{R}_+^{NK}}{\arg\max}\ Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right). \tag{4}$$

Simplifying $Q\left(\boldsymbol{\Lambda},\boldsymbol{\Lambda}^{(r-1)}\right)$ we get,

$$\begin{aligned} &Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right) \\ &= c_K - \frac{1}{2}\sum_{k=1}^{K}\left[\log|\boldsymbol{\Gamma}_k| + \mathsf{Tr}\left\{\boldsymbol{\Gamma}_k^{-1}\mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\}\right\}\right] \\ &\quad - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)\,|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\}, \end{aligned} \tag{5}$$

where $c_K$ is a constant independent of $\boldsymbol{\Lambda}$. We notice that the optimization in the M-step is separable in its variables $\boldsymbol{\Gamma}_k$ and $\boldsymbol{A}$. We get the update of $\boldsymbol{\gamma}_k$ in the M-step as follows (See [30, Appendix D.11] for the detailed derivation):

$$\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}, \tag{6}$$

where we define $\boldsymbol{\mu}_k \triangleq \mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \in \mathbb{R}^N$, and $\boldsymbol{\Sigma}_{(k)} \triangleq \mathbb{E}\left\{(\boldsymbol{x}_k-\boldsymbol{\mu}_k)(\boldsymbol{x}_k-\boldsymbol{\mu}_k)^{\mathsf{T}}|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \in \mathbb{R}^{N\times N}$.

The optimization problem corresponding the dictionary update reduces to

$$\underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ \sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)\Big|\,\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\}. \tag{7}$$

The objective function above can be equivalently written as

$$g(\boldsymbol{A}) = -\mathsf{Tr}\left\{\boldsymbol{M}\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{A}\right\} + \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{A}^{\mathsf{T}}\right\}, \tag{8}$$

where $\boldsymbol{M} \in \mathbb{R}^{N\times K}$ has $\boldsymbol{\mu}_k$ as its $k^{\text{th}}$ column, $\boldsymbol{Y} \in \mathbb{R}^{m\times K}$ has $\boldsymbol{y}_k$ as its $k^{\text{th}}$ column, and the matrix $\boldsymbol{\Sigma} \triangleq \sum_{k=1}^{K}\left(\boldsymbol{\Sigma}_{(k)}+\boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}}\right) \in \mathbb{R}^{N\times N}$. We note that the unit norm constraint on the columns of $\boldsymbol{A}$ combined with the diagonal structure on $\boldsymbol{\Gamma}_k = \mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}\right\}$ implies that the quadratic terms in (7) are independent of $\boldsymbol{A}$, and thus, in (8), the quadratic terms are removed from the objective function.

We note that there is no closed form solution to the quadratic optimization with the unit norm column constraints in (7). Therefore, we solve the optimization problem using two iterative schemes: AM and ALS.

### *A. Alternating Minimization (AM)*

The AM procedure updates one column of $\boldsymbol{A}$ at a time, keeping the other columns fixed. If we fix all columns of $\boldsymbol{A}$ except the $i^{\text{th}}$ column, the optimization problem reduces to

$$\underset{\boldsymbol{A}_i:\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i=1}{\arg\min}\left(\sum_{k=1}^{K}-\boldsymbol{\mu}_k[i]\boldsymbol{y}_k + \sum_{j=1;j\neq i}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j\right)^{\mathsf{T}}\boldsymbol{A}_i. \tag{9}$$

Interestingly, the above reduced optimization problem admits a unique closed form solution provided $\sum_{k=1}^{K}\boldsymbol{\mu}_k[i]\boldsymbol{y}_k - \sum_{j=1;j\neq i}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j \neq \boldsymbol{0}$. If otherwise, we skip the update of that particular column and continue with the update of the next column. Therefore, the dictionary update in the $r^{\text{th}}$ iteration of the EM algorithm reduces to the following recursions for $i = 1,2,\ldots,N$:

$$\boldsymbol{v}_i^{(r,u)} \triangleq \sum_{k=1}^{K}\boldsymbol{\mu}_k[i]\boldsymbol{y}_k - \sum_{j=1}^{i-1}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j^{(r,u)} - \sum_{j=i+1}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j^{(r,u-1)} \tag{10}$$

$$\boldsymbol{A}_i^{(r,u)} = \begin{cases} \frac{1}{\left\|\boldsymbol{v}_i^{(r,u)}\right\|}\boldsymbol{v}_i^{(r,u)} & \text{if } \boldsymbol{v}_i^{(r,u)} \neq \boldsymbol{0} \\ \boldsymbol{A}_i^{(r,u-1)} & \text{otherwise.} \end{cases} \tag{11}$$

where $u$ denotes the AM procedure iteration index. We stop the AM iterations when $\boldsymbol{A}^{(r,u)}$ converges, i.e., its change in successive iterations is small. The pseudo-code for this algorithm, which we call *dictionary learning via SBL (DL-SBL) using AM*, is provided in Algorithm 1.

*Remark:* For the special case when $\boldsymbol{\Sigma}$ is a diagonal matrix and $\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} \neq \boldsymbol{0}$, the optimization problem (7) is separable in the columns of $\boldsymbol{A}$. Then, the AM procedure returns the global minimum of (8) in one iteration.

[3] With a slight abuse of notation, we define $\boldsymbol{\Gamma}_k = \mathsf{Diag}\left\{\boldsymbol{\gamma}_k\right\}$, and not the $k^{\text{th}}$ column of the matrix $\boldsymbol{\Gamma}$.

**Algorithm 1** Dictionary Learning via SBL using AM

```
Input: $\boldsymbol{Y} = \boldsymbol{y}^K$, $N$ and $\sigma^2$
  Parameters: $\epsilon_1$ and $\epsilon_2$ (stopping thresholds)
  Initialize: $r = 0, \boldsymbol{A}^{(0)} = \boldsymbol{1}, \boldsymbol{\gamma}_k^{(0)} = \boldsymbol{1}, k = 1, 2, \ldots, K$
  repeat
    for $k = 1, 2, \ldots, K$ do
      #E-Step:
      $\tilde{\boldsymbol{\Phi}} = \left(\sigma^2 \boldsymbol{I} + \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}}\right)^{-1}$
      $\boldsymbol{\Sigma}_{(k)} = \boldsymbol{\Gamma}_k^{(r)} - \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}} \tilde{\boldsymbol{\Phi}} \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)}$
      $\boldsymbol{\mu}_k = \sigma^{-2} \boldsymbol{\Sigma}_{(k)} \boldsymbol{A}^{(r)\mathsf{T}} \boldsymbol{y}_k$
      $r \leftarrow r + 1$
      #M-Step:
      $\gamma_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}$
    end for
    #Update of $\boldsymbol{A}$ (also part of the M-Step)
    Initialize AM: $u = 0$, $\boldsymbol{A}^{(r,0)} = \boldsymbol{A}^{(r-1)}$
    $\boldsymbol{\Sigma} = \sum_{k=1}^{K} \left(\boldsymbol{\Sigma}_{(k)} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}}\right)$, $\boldsymbol{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]$
    repeat
      $u \leftarrow u + 1$
      for $i = 1, 2, \ldots, N$ do
        $\boldsymbol{v}_i^{(r,u)} = \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \sum_{j=1}^{i-1} \boldsymbol{\Sigma}[i,j] \boldsymbol{A}_j^{(r,u)}$
                $- \sum_{j=i+1}^{N} \boldsymbol{\Sigma}[i,j] \boldsymbol{A}_j^{(r,u-1)}$
        $\boldsymbol{A}_i^{(r,u)} = \begin{cases} \frac{1}{\left\|\boldsymbol{v}_i^{(r,u)}\right\|} \boldsymbol{v}_i^{(r,u)} & \text{if } \boldsymbol{v}_i^{(r,u)} \neq \boldsymbol{0} \\ \boldsymbol{A}_i^{(r,u-1)} & \text{otherwise.} \end{cases}$
      end for
    until $\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\| < \epsilon_2$
    $\boldsymbol{A}^{(r)} = \boldsymbol{A}^{(r,u)}$
  until $\|\boldsymbol{A}^{(r)} - \boldsymbol{A}^{(r-1)}\| + \sum_{k=1}^{K} \|\boldsymbol{\gamma}_k^{(r)} - \boldsymbol{\gamma}_k^{(r-1)}\| < \epsilon_1$
Output: $\{\boldsymbol{\mu}_k, k = 1, 2, \ldots, K\}$ and $\boldsymbol{A}^{(r)}$
```

### B. Armijo Line Search (ALS)

The ALS procedure updates the entire matrix $\boldsymbol{A}$ in every iteration instead of updating one column at a time [31]–[33]. The idea here is to translate the constrained optimization problem into an unconstrained convex optimization problem using Riemannian geometry. The algorithm continuously translates a test point in the opposite direction of the tangent vector at the point, while staying on the manifold, until a reasonable decrease in objective function is obtained, and finally reaches a stationary point. Such a mapping is called a *retraction*, is denoted by $R_{\boldsymbol{A}}$. For Riemannian manifolds, the line search method takes the form

$$\boldsymbol{A}^{(r,u)} = R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right), \tag{12}$$

where $\boldsymbol{Z}^{(r,u-1)}$ is the negative tangent direction of the cost function at $\boldsymbol{A}^{(r,u-1)}$ and $\beta^p\alpha$ is the Armijo step size. The constants $\beta$ and $\alpha$ are the parameters of the algorithm. The step size is chosen so that $p$ is the smallest nonnegative integer that satisfies

$$g\left(R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)\right) - g\left(\boldsymbol{A}^{(r,u-1)}\right) \leq -c\beta^p\alpha \left\|\boldsymbol{Z}^{(r,u-1)}\right\|^2, \tag{13}$$

where the scalar parameter $c \in (0,1)$. The interested readers are referred to [31] for more details on ALS procedure.

We first note that the feasible set $\mathbb{O}$ is the Cartesian product of $N$ unit spheres in $\mathbb{R}^m$ which are submanifolds of the Euclidean space $\mathbb{R}^m$. Since the Cartesian product of Riemannian manifolds is a Riemannian manifold, $\mathbb{O}$ is a Riemannian manifold. We define the Riemannian metric for $\mathbb{O}$ as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathsf{Tr}\left\{\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}\right\}$ for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{O}$. The gradient of the objective function $g$ in the Euclidean space is as follows:

$$\nabla g(\boldsymbol{A}) = -\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} + \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\{\boldsymbol{\Sigma}\}\right). \tag{14}$$

The tangent space of the Cartesian product of manifolds is the Cartesian product of the tangent spaces. Therefore, we get the tangent space as

$$T_{\boldsymbol{A}} = \left\{\boldsymbol{B} : \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{B}_i = 0, \forall i\right\}. \tag{15}$$

The $i^{\text{th}}$ column of the orthogonal projection onto the tangent space is

$$P_{\boldsymbol{A}}(\boldsymbol{Z})_i = \left(\boldsymbol{I} - \boldsymbol{A}_i\boldsymbol{A}_i^{\mathsf{T}}\right)\boldsymbol{Z}_i. \tag{16}$$

Thus, the gradient of the restriction of $g$ to $\mathbb{O}$ is $P_{\boldsymbol{A}}(\nabla g(\boldsymbol{A}))$, and we can choose the $i^{\text{th}}$ column of the retraction as

$$R_{\boldsymbol{A}}(\boldsymbol{Z})_i = \frac{\boldsymbol{A}_i + \boldsymbol{Z}_i}{\|\boldsymbol{A}_i + \boldsymbol{Z}_i\|}. \tag{17}$$

We note that the denominator $\|\boldsymbol{A}_i + \boldsymbol{Z}_i\| \neq 0$ when $\boldsymbol{Z}_i$ is the orthogonal projection onto the tangent space from (16). We call this algorithm DL-SBL using ALS, and summarize its pseudo-code in Algorithm 2.

### C. Comparison of the two optimization procedures

In this subsection, we compare the AM and the ALS procedures to get insights on how to choose between them.

- *Computational complexity:* We assume that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix requires $\mathcal{O}(pqr)$ flops [34]. Each iteration of the AM procedure has a complexity $\mathcal{O}(mKN + mN^2)$. Typically, $K \gg N$ for accurate estimation, and therefore the complexity order is $\mathcal{O}(mKN)$. Thus, the complexity is linear in $m$, $N$ and $K$. On the other hand, the computational complexity of the ALS procedure is also of the order $\mathcal{O}(mKN)$, except for the computation of the step-size parameter $m$. The complexity of this step depends on $c, \beta$ and $\alpha$, and it is hard to determine the precise dependence. However, we have observed in our simulations that the ALS algorithm requires a larger number of iterations and a longer run time to converge compared to the AM procedure for the same initialization. Hence, the AM procedure is faster than the ALS procedure.
- *Memory Requirements:* Both AM and ALS procedures require $\mathcal{O}(N^2)$ sized memory, as the largest matrix we keep track of has size $N \times N$.
- *Parameter tuning:* The AM procedure does not require tuning of any sensitive parameters. However, the ALS procedure has scalar parameters $c, \beta$ and $\alpha$ which determine the rate of convergence, but these parameters do not affect the recovery performance of the overall algorithm.

**Algorithm 2** Dictionary Learning SBL using ALS

```
Input: $\boldsymbol{Y} = \boldsymbol{y}^K$, $N$ and $\sigma^2$
  Parameters: $\epsilon_1$ and $\epsilon_2$ (stopping thresholds)
  Initialize: $r = 0, \boldsymbol{A}^{(0)} = \mathbf{1}, \boldsymbol{\gamma}_k^{(0)} = \mathbf{1}, k = 1, 2, \ldots, K$
  repeat
    for $k = 1, 2, \ldots, K$ do
      #E-Step:
      $\tilde{\boldsymbol{\Phi}} = \left(\sigma^2 \boldsymbol{I} + \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}}\right)^{-1}$
      $\boldsymbol{\Sigma}_{(k)} = \boldsymbol{\Gamma}_k^{(r)} - \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}} \tilde{\boldsymbol{\Phi}} \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)}$
      $\boldsymbol{\mu}_k = \sigma^{-2} \boldsymbol{\Sigma}_{(k)} \boldsymbol{A}^{(r)\mathsf{T}} \boldsymbol{y}_k$
      $r \leftarrow r + 1$
      #M-Step:
      $\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}$
    end for
    #Update of $\boldsymbol{A}$ (also part of the M-Step)
    Initialize ALS: $u = 0,\ \boldsymbol{A}^{(r,0)} = \boldsymbol{A}^{(r-1)}$
    $\boldsymbol{\Sigma} = \sum_{k=1}^{K} \left(\boldsymbol{\Sigma}_{(k)} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}}\right)$, $\boldsymbol{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]$
    repeat
      $u \leftarrow u + 1$
      $\boldsymbol{Z}^{(r,u-1)} = P_{\boldsymbol{A}^{(r,u-1)}} \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r,u-1)} \boldsymbol{\Sigma}\right)$
      Compute the smallest integer $p > 0$ such that
        $g\left(R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)\right) - g\left(\boldsymbol{A}^{(r,u-1)}\right)$
            $\leq -c\beta^p \alpha \left\|\boldsymbol{Z}^{(r,u-1)}\right\|^2$
      $\boldsymbol{A}^{(r,u)} = R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)$
    until $\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\| < \epsilon_2$
    $\boldsymbol{A}^{(r)} = \boldsymbol{A}^{(r,u)}$
  until $\|\boldsymbol{A}^{(r)} - \boldsymbol{A}^{(r-1)}\| + \sum_{k=1}^{K} \|\boldsymbol{\gamma}_k^{(r)} - \boldsymbol{\gamma}_k^{(r-1)}\| < \epsilon_1$
Output: $\{\boldsymbol{\mu}_k, k = 1, 2, \ldots, K\}$ and $\boldsymbol{A}^{(r)}$
```

Hence, the tuning of the parameters of ALS is not very critical. We illustrate this point using experimental results in Section V-A (See Figure 1a, Figure 1b and Table I).

Thus, for practical applications, we prefer AM to ALS as it is computationally less expensive and does not require tuning of any parameters. However, ALS has better theoretical convergence guarantees compared to AM algorithm, which we discuss in Section III.

### D. Comparison with other Bayesian techniques

The main differences between our algorithm and the other Bayesian algorithms in the literature are as follows:

1) Our algorithm does not use Gibbs sampling, unlike the algorithms in [21], [23]. Instead, we use a variational evidence framework which obviates the need for generating posterior samples, and thus our algorithm is faster. Moreover, the ensemble learning based algorithms come with no convergence guarantees. We provide rigorous convergence guarantees for our algorithm in Section III.
2) Our algorithm is similar to the sparse Bayesian dictionary learning with a Gaussian hierarchical model proposed in [23] except for the prior on the dictionary. The algorithm in [23] uses a Gaussian prior on the dictionary elements to obtain a closed form expression for the EM updates. However, the choice of Gaussian prior was heuristically motivated by the fact that the entries of the dictionary are bounded. Since the dictionary is an arbitrary matrix with unit norm columns, the ideal choice of prior on the dictionary columns is a uniform distribution on the unit $m-$dimensional sphere. Hence, we propose to use no prior (which is equivalent to a uniform prior) on the dictionary and learn the dictionary as a deterministic unknown. Due to the better prior model used, our algorithm outperforms the one in [23] in terms of the reconstruction accuracy. The cost paid for this approach is the extra iterative procedure that is nested within the EM algorithm. Using an optimization procedure within the EM framework may appear to be more computationally demanding than an approach with closed form expressions. Nonetheless, from our simulations, we find that our algorithm requires far fewer number of iterations compared to the algorithm in [23]. Hence, the overall run time of the algorithm is much smaller.[4] In other words, the algorithm in this paper is an improved version of Gaussian hierarchical model based SBL algorithm with reduced run time and higher accuracy. We corroborate these arguments through numerical simulations in Section V-B (See Figure 2).
3) Another Bayesian algorithm for dictionary learning is the multimodal sparse Bayesian dictionary learning algorithm [25]. This algorithm is same as the Gaussian hierarchical model based SBL algorithm with a non-informative prior on the dictionary columns, except that it includes an additional projection step. This step projects the columns of the dictionary to the unit norm sphere to avoid instabilities due to the ambiguity in the amplitude. As in the case of the Gaussian hierarchical model based SBL algorithm, this algorithm has a closed form expression for the M-step. As explained above, the algorithm effectively uses a non-informative prior on the dictionary atoms instead of using a uniform distribution on the $m-$dimensional unit sphere. Further, the convergence guarantees in [25] do not apply to the algorithm that involves the projection step, which is crucial to the success of the algorithm. Since our cost function is carefully designed to handle the amplitude ambiguity, our algorithm outperforms the multimodal sparse Bayesian dictionary learning algorithm. We illustrate this through numerical simulations in Section V-B (See Figure 2).

## III. Convergence Analysis of Optimization Procedures

In this section, we discuss the convergence properties of the AM and ALS procedures proposed to solve (7).

[4] A similar observation can be found, in the context of sparse signal recovery, in [35]. Iterative reweighted $\ell_2$ algorithms are typically slower than iterative reweighted $\ell_1$ algorithms, even though the former admits closed form expressions in the iterations.

**Proposition 1** (Function value convergence)**.** *The sequences of cost function values* $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ *generated by the AM and the ALS procedures are non-increasing and convergent.*

*Proof.* See Appendix A. □

While above proposition guarantees that the cost function value converges, it does not establish the convergence of the iterates. Hence, we study the convergence behavior of the iterates in the next subsections. Before presenting the results, we start with a definition that applies to both the AM and ALS procedures.

**Definition 1** (Nash equilibrium)**.** *The matrix* $\boldsymbol{A}$ *with unit norm columns is said to be a Nash equilibrium point of* (7) *if*

$$g(\boldsymbol{A}) \leq g\left([\boldsymbol{A}_1,\ldots,\boldsymbol{A}_{i-1},\boldsymbol{a},\boldsymbol{A}_{i+1},\ldots,\boldsymbol{A}_N]\right), \tag{18}$$

*for any unit-norm vector* $\boldsymbol{a}$ *and for* $i=1,2,\ldots,N$*.*

Every column of a Nash equilibrium is optimal when other columns of the dictionary are held fixed, that is, one cannot unilaterally improve the cost function in (7) by updating any single column. We now proceed with our analysis of the convergence of the AM procedure in the next subsection.

### A. AM Procedure

The iterative AM procedure can be viewed as a fixed point iteration with the update mapping dictated by the function whose stationary point is sought. The following result shows that the fixed points of the updates generated by the AM procedure are Nash equilibria of (7).

**Proposition 2** (Nash Equlibrium)**.** *Let* $G:\mathbb{O}\to\mathbb{O}$ *be the update mapping of AM procedure, i.e.,* $\boldsymbol{A}^{(r,u+1)}=G(\boldsymbol{A}^{(r,u)})$*. Then, a matrix* $\boldsymbol{A}^*$ *is a fixed point of* $G$ *if and only if* $\boldsymbol{A}^*$ *is a Nash equilibrium point of* (7)*. Further, all Nash equilibrium points are stationary points of the cost function.*

*Proof.* See Appendix B. □

**Corollary 1.** *A matrix* $\boldsymbol{A}$ *with unit norm columns is a Nash equilibrium point of the objective function in* (7) *if and only if* $\boldsymbol{A}$ *satisfies the relation:*

$$\boldsymbol{A}\boldsymbol{L}=\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}-\boldsymbol{A}\left(\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right) \tag{19}$$

*for some diagonal positive semidefinite (psd) matrix* $\boldsymbol{L}$*.*

*Proof.* The result directly follows from the form of the fixed points shown in the proof of Proposition 2. □

We note that the update mapping of the AM procedure does not have a closed form expression owing to the sequential, column-wise update of the dictionary. Due to this, although the above theorem characterizes its fixed points, it is hard to establish the convergence of the iterates. On the other hand, it is possible to show several interesting convergence properties of the iterates in the ALS procedure. We discuss this next.

### B. ALS Procedure

We begin by noting that establishing convergence guarantees for the ALS procedure is challenging because the optimization problem in (7) is non-convex in $\boldsymbol{A}$. In particular, since $\boldsymbol{A}$ is constrained to lie in the set $\mathbb{O}$, the set of all matrices with unit-norm columns, establishing convergence requires analyzing the convergence behavior over Riemannian manifolds. Existing results in this direction, e.g., [36]–[40], consider convex optimization problems, and very few results are known for the non-convex case. In [41], the authors studied the convergence of a proximal algorithm applied to nonsmooth functions that satisfy the Łojasiewicz inequality around their generalized stationary points. Based on this, convergence of iterative solvers for quadratic optimization of a matrix valued variable over the space of orthogonal matrices was shown in [42]. In [43], quadratic optimization over the space of unit norm vectors was studied. These results, when extended to a matrix setting, lead to a unit norm constraint on the rows of the matrix, and hence are not applicable in our case. The convergence of an ALS type procedure for a quadratic optimization problem under *unit-norm column* constraints has not been studied in the literature, and requires new analysis.

To discuss the convergence properties of the ALS procedures, we consider an equivalent unconstrained version of the optimization problem in (7) as follows:

$$\arg\min_{\boldsymbol{A}}\ \mathsf{Tr}\left\{-\boldsymbol{M}\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{A}+\frac{1}{2}\left(\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right\}+\delta_{\text{norm}}(\boldsymbol{A}). \tag{20}$$

Here, we define $\delta_{\text{norm}}$ as a barrier function corresponding to the feasible region of (7):

$$\delta_{\text{norm}}(\boldsymbol{A}) \triangleq \begin{cases} 0, & \text{if } \boldsymbol{A}\in\mathbb{O} \\ \infty, & \text{otherwise.} \end{cases} \tag{21}$$

Also, let $\tilde{g}:\mathbb{R}^{m\times N}\to\mathbb{R}$ denote the objective function of (20). The stationary points of (7) are the points where the subgradient of $\tilde{g}$ vanishes.[5]

**Theorem 1** (Convergence of iterates)**.** *The sequence output by the ALS procedure,* $\{\boldsymbol{A}^{(r,u)}\}_{u\in\mathbb{N}}$*, is globally convergent.*

*Proof.* See Appendix C. □

The above theorem guarantees that the iterates of the ALS procedure converge irrespective of the initial point. However, it does not ensure that the algorithm converges to the same point for any initial point. Such a guarantee exists only if the cost function has only one limit point. Hence, we next characterize the limits points of the sequence of iterates.

**Proposition 3** (Characterization of limits)**.** *The limit* $\boldsymbol{A}^{(r)}$ *of the sequence* $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ *generated by the ALS procedure satisfies the relation:*

$$\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}-\boldsymbol{A}^{(r)}\left(\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)=\boldsymbol{A}^{(r)}\boldsymbol{L}, \tag{22}$$

*for some diagonal matrix* $\boldsymbol{L}$*. Moreover,*

[5] We note that we use an extended definition of sub-gradient as the function $\tilde{g}$ is non-convex.

*1) $\boldsymbol{A}^{(r)}$ is a Nash equilibrium point of* (7) *if and only if $\boldsymbol{L}$ is a positive semidefinite matrix.*

*2) $\boldsymbol{A}^{(r)}$ is a local minimum if and only if $\boldsymbol{L}+\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is a positive semidefinite matrix. Further, $\boldsymbol{A}^{(r)}$ is a strict local minimum if and only if $\boldsymbol{L}+\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is a positive definite matrix.*

*Proof.* See Appendix D. □

We make the following observations from the above results:

- As in the case of the AM procedure, the update mapping of ALS is not available in closed form because of the step size selection process. However, the results characterize the fixed points of the mapping.
- The initialization $\boldsymbol{A}^{(r,0)}$ need not be a feasible point of (7). Because of the retraction step which projects the iterates to the feasible set, the algorithm can be initialized from any bounded matrix.
- The results are independent of the estimates from the outer iteration loop of the EM algorithm and the dimension of the dictionary. Thus, the results are applicable to any quadratic cost function of the form (7).
- Given $\boldsymbol{A}^{(r)}, \boldsymbol{M}, \boldsymbol{Y}$ and $\boldsymbol{\Sigma}$, the conditions for the Nash equilibrium and local minimum are easily verifiable.

Now, for any first order method such as the ALS procedure, the best guarantees one can obtain are that it converges to a stationary point. Further, we can determine whether the stationary point is a local minimum using the test in step 2 of Proposition 3. Beyond this, the only guarantee one can provide for first order methods is that of stability of the limit points. Stability implies that the algorithm converges to a limit point whenever it is initialized close enough to it. Formally, we define stability as follows:

**Definition 2** (Stability)**.** *Let $G:\mathbb{O}\rightarrow\mathbb{O}$ be the update mapping of an iterative algorithm, i.e., $\boldsymbol{A}^{(r,u+1)}=G(\boldsymbol{A}^{(r,u)})$. Also, we let $G^{(u)}(\cdot)$ denote the result of $u$ applications of $G$:*

$$G^{(1)}(\boldsymbol{A})=G(\boldsymbol{A});\quad G^{(u+1)}(\boldsymbol{A})=G\left(G^{(u)}(\boldsymbol{A})\right). \tag{23}$$

*The matrix $\boldsymbol{A}^*$ said to be a stable point of the iterative algorithm if, for every neighborhood $\mathcal{U}$ of $\boldsymbol{A}^*$, there exists a neighborhood $\mathcal{V}$ of $\boldsymbol{A}^*$ such that, for all $\boldsymbol{A}\in\mathcal{V}$ and any positive integer $u$, it holds that $G^{(u)}(\boldsymbol{A})\in\mathcal{U}$.*

We have the following characterization of the stability of the fixed points of the ALS procedure, based on whether the fixed point is a local minimum or not.

**Theorem 2** (Stability)**.** *Let $\boldsymbol{A}^{(r)}$ be a limit point of the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ generated by the ALS procedure. Then,*

*(i) If $\boldsymbol{A}^{(r)}$ is not a local minimum of $\tilde{g}$, then $\boldsymbol{A}^{(r)}$ is not a stable point of the ALS procedure.*

*(ii) If $\boldsymbol{A}^{(r)}$ is a strict local minimum of $\tilde{g}$, then the algorithm converges to $\boldsymbol{A}^{(r)}$ if the initial point $\boldsymbol{A}^{(r,0)}$ is sufficiently close to $\boldsymbol{A}^{(r)}$.*

*Proof.* See Appendix E. □

An implication of Theorem 2 is that the ALS procedure converges to a local minimum of the cost function, unless the initial condition is carefully constructed to be adversarial in nature. Also, as in the previous case, the results are independent of the estimates from the outer iteration loop of the EM algorithm and the dimension of the dictionary. Thus, Theorem 2 is applicable to any optimization of the form (7).

In this section, we analyzed the convergence properties of the inner loop in the M-step of EM algorithm. Our analysis guarantees that the optimization procedure has good converge properties. As a consequence, and by virtue of the well-known properties of the EM algorithm, DL-SBL is globally convergent. Next, we formally prove the convergence of the overall DL-SBL algorithm and analyze the minima of the DL-SBL cost function given by (3).

## IV. Analysis of DL-SBL Algorithm

The DL-SBL algorithm is not an EM algorithm in the strict sense because the M-step of the DL-SBL is not guaranteed to converge to the global minimizer, unlike the conventional EM. However, DL-SBL inherits many good properties of EM such as a monotonic reduction of the cost function. In this section, we build on the results in Section III and study the characteristics of the DL-SBL algorithm and the cost function.

### A. Convergence of DL-SBL

We start by stating the following result, which asserts that the DL-SBL cost converges.

**Proposition 4.** *Suppose that $\sigma^2>0$. The sequence of cost function values $\left\{T(\boldsymbol{\Lambda}^{(r)})\right\}_{r\in\mathbb{N}}$ generated by the DL-SBL algorithm via ALS procedure converges to $T(\boldsymbol{\Lambda}^*)$ for some $\boldsymbol{\Lambda}^*\in\mathbb{O}\times\mathbb{R}_+^{KN}$.*

*Proof.* See Appendix F. □

Next, we characterize the properties of the iterates generated by the algorithm.

**Theorem 3.** *Suppose that $\sigma^2>0$. The iterates $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ of the outer loop of the DL-SBL algorithm converge to the set of stationary points of the DL-SBL cost function given by* (3)*. Moreover, if a limit point $\boldsymbol{\Lambda}^*$ of the sequence $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ is not a local minimum of $T$, then $\boldsymbol{\Lambda}^*$ is not a stable point of the ALS procedure.*

*Proof.* See Appendix G. □

The above results guarantee that the cost function values $\left\{T(\boldsymbol{\Lambda}^{(r)})\right\}$ converge to $T(\boldsymbol{\Lambda}^*)$ for some stationary point $\boldsymbol{\Lambda}^*$. They also guarantee that the sequence of iterates converges to a compact and connected subset of a level set of the cost function, although it does not necessarily converge to a single point. Theorem 3 also gives insights to the stability of the fixed points of the algorithm, similar to Theorem 2. Further, as in the case of the results in Section III, the above results hold for any values of system dimensions: $m, N$, and $K$, and sparsity level $s$.

The next question that we address is on how good the final solution of DL-SBL is, by analyzing the minima of the DL-SBL cost function given by (3).

### *B. Analysis of Minima of the Cost Function*

First, note that, in the context of dictionary learning, the problem of finding the sparse representation of a given set of vectors $\boldsymbol{y}^K$, uniqueness of the solution is defined up to an unavoidable permutation of the unit-norm columns of $\boldsymbol{A}$ and rows of $\boldsymbol{X}$, where $\boldsymbol{X} \in \mathbb{R}^{N \times K}$ is the matrix obtained by stacking the sparse vectors $\boldsymbol{x}_k$. We now present necessary conditions for the uniqueness of the solution:

**Proposition 5.** *Consider the dictionary learning problem under noiseless condition $\sigma^2 = 0$, i.e., for any given $\boldsymbol{Y}$, the problem of finding matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ such that $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, the columns of $\boldsymbol{A}$ have unit norm and the columns of $\boldsymbol{X}$ have at most $s$ non-zero entries. The solution to the problem is unique only if the following conditions are satisfied:*

$$\mathsf{Rank}\left\{\boldsymbol{X}\right\} = N \tag{24}$$

$$\mathsf{Rank}\left\{\boldsymbol{A}_{\mathcal{S}_k}\right\} = |\mathcal{S}_k| < m, \tag{25}$$

*where $\mathcal{S}_k$ is the support of $\boldsymbol{x}_k$ and $\boldsymbol{A}_{\mathcal{S}_k} \in \mathbb{R}^{m \times |\mathcal{S}_k|}$ is the submatrix of $\boldsymbol{A}$ formed by the columns indexed by $\mathcal{S}_k$. Further, if $\max_{k=1,2,\ldots,K} \|\boldsymbol{x}_k\|_0 = 1$, the conditions are sufficient.*

*Proof.* See Appendix H. □

We note that the necessary conditions required to ensure the uniqueness of the solution of the dictionary learning problem is applicable for any dictionary learning algorithm, and in particular, DL-SBL. Next, we establish that the cost function in (3), when minimized, has the desired global minima.

**Theorem 4.** *Suppose the tuple $(\boldsymbol{A}^*, \boldsymbol{X}^*)$ satisfies the necessary conditions* (24) *and* (25)*. Also, let $\left\{\boldsymbol{\Gamma}_k^* \in \mathbb{R}^{N \times N}\right\}_{k=1}^K$ be a set of nonnegative diagonal matrices denoting the covariance matrix of the sparse vectors such that*

$$\boldsymbol{x}_k^* = \boldsymbol{\Gamma}_k^{*1/2}\left(\boldsymbol{A}^*\boldsymbol{\Gamma}_k^{*1/2}\right)^\dagger \boldsymbol{y}_k \tag{26}$$

$$\text{and } 0 < c < \min_{k=1,2,\ldots K} \gamma_k^* \tag{27}$$

*where $\gamma_k^*$ is the smallest nonzero entry of $\boldsymbol{\Gamma}_k^*$ and $c$ is a universal constant. Then, as the noise variance $\sigma^2 \to 0$, the global minimum of* (3) *is achieved at $\left(\boldsymbol{A}^*\boldsymbol{P}, \left\{\boldsymbol{P}\boldsymbol{\Gamma}_k^*\boldsymbol{P}\right\}_{k=1}^K\right)$ where $\boldsymbol{P}$ is a signed permutation matrix.*

*Proof.* See Appendix I. □

We note that the sparsest solution of (3) is $(\boldsymbol{A}^*, \boldsymbol{X}^*)$ due to (25). Although we assume that the necessary conditions (24) and (25) hold, the theorem holds under the mild condition that

$$\max_{k=1,2,\ldots,K} \|\boldsymbol{x}_k\|_0 < m. \tag{28}$$

However, under the above condition, uniqueness is not guaranteed, i.e., solutions with suboptimal sparsity may also globally minimize the cost function.

We know that the DL problem is NP-hard [44]. Thus, it is not surprising that the cost function obtained using SBL framework may have multiple local minima. Nonetheless, extending the results of the original SBL algorithm on sparse recovery [29], we can show that all the local minima of the function are achieved at sparse solutions.

Table I
COMPARISON OF ALS CONVERGENCE BEHAVIOUR WITH VARYING STEP SIZE PARAMETERS $\beta$ AND $\alpha$

| **Setting** | | **Fit parameters** | | **no. of iterations** | **run time (s)** |
|---|---|---|---|---|---|
| | | a | b | | |
| $\alpha = 0.1$ | $\beta = 0.01$ | -0.034 | -0.093 | 565.04 | 1.33 |
| | $\beta = 0.1$ | -0.036 | -1.102 | 490.09 | 1.5 |
| | $\beta = 0.9$ | -0.044 | -1.554 | 480.63 | 13.68 |
| $\beta = 0.1$ | $\alpha = 0.01$ | -0.036 | -1.118 | 494.26 | 1.55 |
| | $\alpha = 0.1$ | -0.036 | -1.102 | 490.09 | 1.50 |
| | $\alpha = 0.9$ | -0.037 | -0.226 | 486.60 | 1.51 |

Table II
COMPARISON OF ALS AND AM CONVERGENCE BEHAVIOR

| **Algo.** | **Fit parameters** | | **no. of iterations** | **run time (s)** |
|---|---|---|---|---|
| | a | b | | |
| AM | -0.0427 | -0.4603 | 248.95 | 0.5828 |
| ALS | -0.0361 | -1.1022 | 490.09 | 1.5020 |

**Theorem 5.** *Every $\gamma_k$ corresponding to the local minimum of* (3) *is at most $m-$sparse, regardless of the value of noise variance $\sigma^2$.*

*Proof.* See Appendix J. □

## V. SIMULATION RESULTS

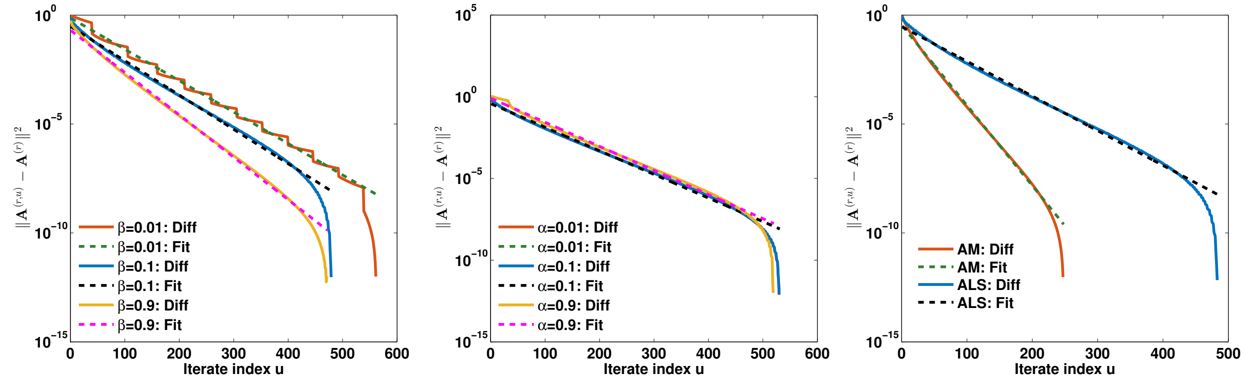
We use the following simulation setup to evaluate the performance of the algorithms and validate the theoretical convergence results in Section V-A and Section V-B. The locations of nonzero coefficients are chosen uniformly at random, and the nonzero entries are independent and identically Gaussian distributed with zero mean and unit variance. The length of measurement vector is chosen as $m = 20$ and SNR $= 20$ dB. The columns of dictionary matrix $\boldsymbol{A}$ are drawn uniformly from the surface of the $m$-dimensional unit hypersphere [45].

### *A. Convergence*

To study the convergence of the AM procedure, we take size of training data set as $K = 1000$. We generate sparse signals of length $N = 60$, each with $s = 6$ nonzero entries. We look at the first iteration ($r = 1$) of the EM algorithm because that requires the maximum number of inner iterations to converge, and thus illustrates the convergence behavior well.

Figure 1 shows the $\ell_2$ squared norm of the difference between the iterates and the limit point, given by $\|\boldsymbol{A}^{(1,u)} - \boldsymbol{A}^{(1)}\|^2$, of the AM and ALS procedures under different settings. We set $\beta = \alpha = 0.1$ for Figure 1, unless specified otherwise. The curves labeled `Diff` and `Fit` correspond to the curves obtained via numerical experiments and by fitting the function $f(u) = \exp(au + b)$ on the values, respectively, where $a < 0$ and $b$ are parameters of the curve. The values of the parameters averaged over 100 experiments are listed in Table I and Table II. Our observations are as follows:

- *Rate of convergence:* From Figure 1, we see that the curve is well approximated using an exponential function



(a) Varying value of $\beta$

(b) Varying value of $\alpha$

(c) Comparison of AM and ALS

Figure 1. Convergence of ALS procedure ((a), (b)) and comparison with AM (c), with $K = 1000$, $m = 20$, $N = 60$, $s = 6$, and SNR = 20 dB, for the first iteration of EM algorithm.

for moderate values of iteration number. Further, the tail of the curve exhibits a faster-than-exponential decay. Interestingly, both AM and ALS procedures exhibit the same behavior for all choices of $\beta$ and $\alpha$.

- *Step size parameter $\beta$:* In the backtracking step of the ALS procedure, we need to evaluate the optimal step size by searching over the step sizes $\beta\alpha, \beta^2\alpha, \beta^3\alpha, \ldots$. For smaller $\beta$, the number of iterations to reach the stopping threshold decreases, as the search domain is larger. However, as $\beta$ increases, the optimal value of $p$ also increases, which results in a higher run time as it requires $p$ function evaluations and comparisons. Thus, using $\beta \leq 0.1$ strikes a good balance between run time and the number of iterations.
- *Step size parameter $\alpha$:* For the ALS procedure, $\alpha$ does not have any effect on rate of convergence or run time. This is because, the size of the discrete search in the backtracking step does not depend on the value of $\alpha$.
- *Comparison of AM and ALS:* From Figure 1c and Table II, we see that the AM algorithm converges faster than ALS and requires fewer number of iterations for the same stopping threshold. Therefore, the AM procedure is computationally more attractive than the ALS in practice, although the ALS procedure comes with stronger theoretical convergence guarantees.

*Remark:* The number of iterations required by the procedure dramatically reduces as $r$ increases. All the plots shown here correspond to $r = 1$. However, for $r > 10$, only about 2-4 iterations are required for the inner optimization, making it computationally very efficient.

### *B. Performance of the Algorithms*

In this subsection, we compare the performance of our algorithms with other popular algorithms in literature. Here, we do not show separate curves for DL-SBL using the ALS and AM algorithms, as their performances are virtually identical.

For fairness of comparison, the noise level information is provided to all algorithms. For SimCo, KSVD and MOD, it is used to set the error threshold in the orthogonal matching pursuit (OMP) step of the algorithm; the threshold is set to be $1.15$ times the noise variance. For DL-SBL, GAMP, Gaussian hierarchical model based SBL, multimodal sparse Bayesian dictionary learning, and Bayesian KSVD, the noise variance is an input to the algorithm. We use the version of Gaussian hierarchical model based SBL and Bayesian KSVD which do not learn the noise level, but take the noise level as an input.

*1) Synthetic Data:* We use the same setup as in [15]. We generate sparse signals of length $N = 50$, each with $s = 3$ nonzero entries. We let $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ denote the estimate and true value of the sparse vector, respectively, and $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}$ denote the estimate and true value of the dictionary, respectively. We use the following metrics evaluating the performance.

(i) Dictionary recovery success rate (DRSR) [15], which is the fraction of successfully recovered columns of the dictionary. A column is said to be successfully recovered if the magnitude inner product between the column in the true dictionary and any of the estimated dictionary columns exceeds $0.99$.

(ii) Relative distortion (RD) [17], defined as:

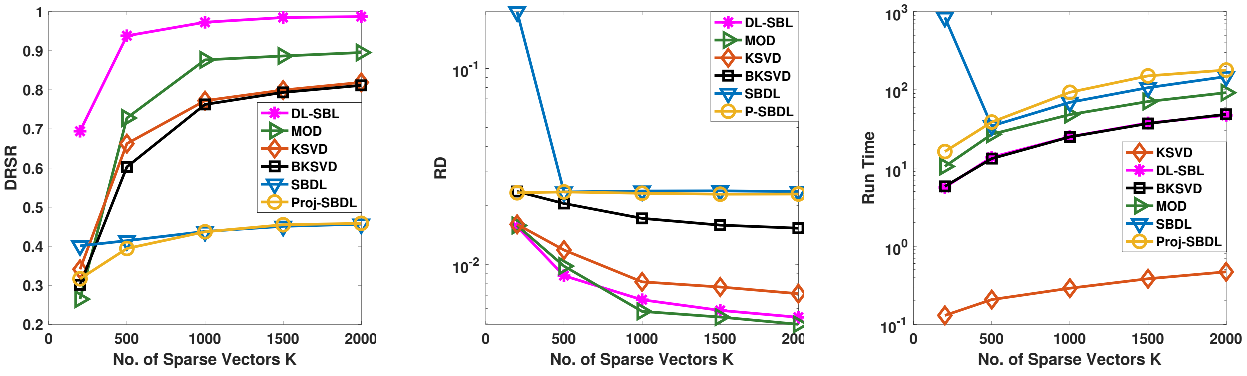
$$\text{RD} \triangleq \frac{\sum_{k=1}^{K} \|\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_k - \boldsymbol{A}\boldsymbol{x}_k\|^2}{\sum_{k=1}^{K} \|\boldsymbol{A}\boldsymbol{x}_k\|^2}. \qquad (29)$$

(iii) Run time, which is the time required to complete the computations. It measures the computational complexity.

We refer to the DRSR and RD metrics jointly as the recovery performance of the algorithm. Note that, any solution of the form $\{\boldsymbol{AP}, \boldsymbol{Px}_k, k = 1, 2, \ldots, K\}$, where $\boldsymbol{P}$ is a signed permutation matrix[6] is a solution to the dictionary learning problem. Consequently, the error metric $\frac{\|\boldsymbol{A} - \hat{\boldsymbol{A}}\|^2}{\|\boldsymbol{A}\|^2}$ does not account for the inherent non-uniqueness of the solution. Hence, we use DRSR as a measure of how well the dictionary is recovered, and RD is a measure of how well the recovered solution matches with the measurements.

Figure 2 compares the proposed algorithm with the following algorithms:

[6] A matrix is said to be a signed permutation matrix if it has exactly one nonzero entry which is either 1 or $-1$ in each row and each column.



(a) (b) (c)

Figure 2. Comparison of DL-SBL with KSVD, MOD, Gaussian hierarchical model based SBL algorithm, multimodal sparse Bayesian dictionary learning, and Bayesian KSVD, when the number of input vectors is varied. The performance of DL-SBL is superior to the other three algorithms.

- KSVD [15]
- MOD [14]
- Gaussian hierarchical model based SBL algorithms [23] (labeled as `SBDL`)
- Multimodal sparse Bayesian dictionary learning [25], [26] (labeled as `Proj-SBDL`)
- Bayesian KSVD [27].

For the Gaussian hierarchical model based SBL, using a non-informative prior results in the best performance. Therefore, we use that version of the algorithm for comparison.

The performance of all the algorithms improve with $K$, as more information about the dictionary is available to the algorithm. The DL-SBL algorithm outperforms the other algorithms in terms of both DRSR and RD. The run time demanded by our algorithm is larger than K-SVD, but it is lower than the other two algorithms.

The Gaussian hierarchical model based SBL and multimodal sparse Bayesian dictionary learning have similar performance except for $K = 200$. When the number of measurements is very small ($K = 200$), the Gaussian hierarchical model based SBL algorithm fails to converge, resulting in higher run time and poor performance. The projection step used in the multimodal sparse Bayesian dictionary learning eliminates such instabilities. However, in the regime shown in Figure 2, the performance of both the algorithms is inferior to the other algorithms in the literature. This observation agrees with the intuitive explanation presented in Section II-D that the Gaussian hierarchical model based SBL algorithm requires a larger number of measurements compared to the DL-SBL algorithm to achieve good performance.

Table III
COMPARISON OF PSNR VALUES OF DIFFERENT ALGORITHMS WITH VARYING NOISE VARIANCE

| Noise Standard Deviation | 5 | 10 | 15 | 25 |
|---|---|---|---|---|
| SimCo | 38.9843 | 33.7205 | 30.8103 | 27.3856 |
| KSVD | **39.0861** | 33.8418 | 30.8928 | 27.3751 |
| MOD | 38.8720 | 33.8818 | **31.0586** | 27.5354 |
| DL-SBL | 39.0680 | **33.9115** | 31.0513 | **27.6371** |
| GAMP | 38.7975 | 33.7574 | 30.9353 | 27.4408 |
| BKSVD | 39.0317 | 33.8861 | 31.0124 | 27.6041 |

Table IV
COMPARISON OF SSIM VALUES OF DIFFERENT ALGORITHMS WITH VARYING NOISE VARIANCE

| Noise Standard Deviation | 5 | 10 | 15 | 25 |
|---|---|---|---|---|
| SimCo | 0.9643 | 0.8936 | 0.8289 | 0.7396 |
| KSVD | 0.9648 | 0.8946 | 0.8297 | 0.7393 |
| MOD | 0.9646 | **0.8959** | **0.8324** | 0.7425 |
| DL-SBL | **0.9650** | 0.8958 | 0.8320 | **0.7440** |
| GAMP | 0.9600 | 0.8876 | 0.8252 | 0.7384 |
| BKSVD | 0.9644 | 0.8953 | 0.8317 | 0.7439 |

*2) Image Denoising:* We next consider the application of DL to the problem of image denoising. Here, the goal is to remove zero-mean white and homogeneous Gaussian additive noise from a given image. We adopt the same simulation setup as in [15], and use 10 randomly chosen gray scale images from the Berkeley segmentation database. The noise standard deviations used in this benchmark are 5, 10, 15, and 25 gray levels. For every image, we learn the dictionary using $K = 6000$ uniformly randomly chosen blocks of size $m = 8 \times 8 = 64$ pixels. The length of the sparse vectors $N$ is taken as 256.

For all the algorithms, once the dictionary is learned, the complete image is reconstructed using the OMP algorithm with the corrupted image and the learned dictionary as inputs and error threshold as 1.15 times the noise variance. We reconstruct the image as $8 \times 8$ overlapping blocks which are then combined by averaging the overlapping pixels. The peak SNR (PSNR) and structural similarity index (SSIM) values of the images reconstructed by several algorithms are shown in Table III and Table IV, respectively. The tables show the median values of the corresponding measures for each noise levels. The following algorithms are compared:

- Simultaneous codeword optimization (SimCo) [17]
- K-singular value decomposition (K-SVD) [15]
- Method of optimal directions (MOD) [14]
- Bilinear generalized approximate message passing algorithm (GAMP) [22]
- Bayesian K-SVD (BKSVD) [27].

The results show that the performance of DL-SBL matches that of the other algorithms at all noise levels, and it offers the best performance at a noise level of 25. At smaller noise levels (5 and 10), there is no clear winner as the best PSNR value and the best SSIM value correspond to different algorithms including DL-SBL. At noise level 15, MOD has the best performance. However, the performance of DL-SBL is close to the best performing algorithm for both metrics for all noise levels. Therefore, the performance of our algorithm is similar to the state-of-the-art algorithms.

## VI. Conclusions

In this paper, we analyzed a Bayesian algorithm for jointly recovering a dictionary matrix and a set of sparse vectors from a training set containing noisy underdetermined linear measurements. We developed the algorithm using the SBL framework, and implemented it using the EM algorithm, with the dictionary matrix and the variances of the entries of the sparse vectors as unknown parameters. The EM algorithm requires one to solve a non-convex optimization problem in the M-step, which we tackled using an AM or ALS procedure. We compared the AM and ALS procedures in terms of their computational complexity and memory requirements. We also provided a rigorous convergence analysis of the proposed optimization procedures. Further, by direct analysis of the cost function involved, we showed that the DL-SBL algorithm is likely to output the sparsest representation of the input vectors. We empirically showed the efficacy of our algorithm compared to existing algorithms, when applied to the image denoising problem. Designing low complexity DL algorithms in an online setup where the data is sequentially available is an interesting direction for future work.

## Acknowledgements

We gratefully thank Rajiv Soundararajan for helpful discussions on the image processing application presented in this work. We also thank the anonymous reviewers, whose insightful feedback helped improve the paper.

## Appendix A
## Proof of Proposition 1

*Proof.* For the AM procedure, since we optimize one column of $\boldsymbol{A}$ at a time, it is easy to see that

$$g\left(\boldsymbol{A}^{(r,u-1)}\right) \geq g\left(\boldsymbol{A}^{(r,u)}\right). \tag{30}$$

The above relation holds even if we skip the update of a column when $\left\|\boldsymbol{v}_i^{(r,u)}\right\| = 0$, in which case the value of the cost function remains unchanged. Similarly, from (13), the sequence $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ generated by the ALS algorithm is also nonincreasing. Thus, in both cases, $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ is a nonincreasing sequence bounded by $g\left(\boldsymbol{A}^{(r,0)}\right)$ from above. From (8), we have

$$\begin{aligned} g(\boldsymbol{A}) &= \frac{1}{2}\mathsf{Tr}\left\{\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\right)^{\mathsf{T}}\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\right) + \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathsf{T}}\right\} \\ &\qquad - \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{Y} + \boldsymbol{\Sigma}\right\} - N/2 \quad (31) \\ &\geq -\frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{Y} + \boldsymbol{\Sigma}\right\} - N/2. \quad (32) \end{aligned}$$

Therefore, the nonincreasing sequence $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ is bounded from below, and hence it converges. □

## Appendix B
## Proof of Proposition 2

*Proof.* The first part of the result directly follows from the properties of AM. Further, any stationary point of the cost function takes the following form:

$$\boldsymbol{A}\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right) \tag{33}$$

for some diagonal matrix $\boldsymbol{L}$. From (11), we get

$$G(\boldsymbol{A})_i \left\|\boldsymbol{v}_i\right\| = \boldsymbol{v}_i, \tag{34}$$

where

$$\begin{aligned} \boldsymbol{v}_i &= \sum_{k=1}^{K}\boldsymbol{\mu}_k[i]\boldsymbol{y}_k - \sum_{j=1}^{i-1}\boldsymbol{\Sigma}[i,j]G(\boldsymbol{A})_j - \sum_{j=i+1}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j \\ &= \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \boldsymbol{G}(\boldsymbol{A})\left(\hat{\boldsymbol{\Sigma}}^{\mathsf{T}}\right)_i - \boldsymbol{A}\hat{\boldsymbol{\Sigma}}_i, \quad (35) \end{aligned}$$

where $\hat{\boldsymbol{\Sigma}}$ is lower triangular with zero diagonal entries and $\hat{\boldsymbol{\Sigma}} + \hat{\boldsymbol{\Sigma}}^{\mathsf{T}} = \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$. When $\boldsymbol{A}$ is a fixed point of $G$, we get

$$\boldsymbol{v}_i = \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)_i. \tag{36}$$

Now, from (34) and (36), it can be seen that $\boldsymbol{A}$ satisfies (33) with $\boldsymbol{L}_{ii} = \left\|\boldsymbol{v}_i\right\| \geq 0$, which concludes the proof. □

## Appendix C
## Proof of Theorem 1

The proof of the theorem rests on the following lemmas.

**Lemma 1.** *Let $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ be a sequence generated by the ALS procedure. Then, there exists $C_1 > 0$ such that*

$$\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r,u)}\right) \geq C_1\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2. \tag{37}$$

*Proof.* We note from (17) that

$$\boldsymbol{A}_i^{(r,u)} = \frac{\boldsymbol{A}_i^{(r,u-1)} + \beta^p\alpha\boldsymbol{Z}_i^{(r,u-1)}}{\left\|\boldsymbol{A}_i^{(r,u-1)} + \beta^p\alpha\boldsymbol{Z}_i^{(r,u-1)}\right\|}. \tag{38}$$

Also, from (16), we know that

$$\boldsymbol{A}_i^{(r,u-1)\mathsf{T}}\boldsymbol{Z}_i^{(r,u-1)} = 0. \tag{39}$$

Therefore, we get

$$\begin{aligned}
\frac{1}{2}\left\|\boldsymbol{A}^{(r,u-1)}-\boldsymbol{A}^{(r,u)}\right\|^{2}&\\
&=\sum_{i=1}^{N}\frac{1}{2}\left\|\boldsymbol{A}_{i}^{(r,u-1)}-\boldsymbol{A}_{i}^{(r,u)}\right\|^{2} \qquad (40)\\
&=\sum_{i=1}^{N}\left(1-\boldsymbol{A}_{i}^{(r,u-1)\mathsf{T}}\boldsymbol{A}_{i}^{(r,u)}\right) \qquad (41)\\
&=\sum_{i=1}^{N}\left(1-\frac{1}{\sqrt{1+\left\|\beta^{p}\alpha\boldsymbol{Z}_{i}^{(r,u-1)}\right\|^{2}}}\right) \qquad (42)\\
&\leq\sum_{i=1}^{N}\left\|\beta^{p}\alpha\boldsymbol{Z}_{i}^{(r,u-1)}\right\|^{2}, \qquad (43)\\
&\leq\frac{1}{c}\left[g\left(\boldsymbol{A}^{(r,u-1)}\right)-g\left(\boldsymbol{A}^{(r,u)}\right)\right] \qquad (44)
\end{aligned}$$

where (41) is because $\boldsymbol{A}_i^{(r,u-1)}$ and $\boldsymbol{A}_i^{(r,u)}$ are unit norm vectors, and (42) is a consequence of (38) and (39), (43) is due to the fact that $x^2+1/\sqrt{1+x^2}-1\geq 0$ for all $x\in\mathbb{R}$, and (44) follows from (13). Thus, the proof is complete. □

**Lemma 2** (Subgradient of $\delta_{\text{norm}}$)**.** *For any $\boldsymbol{A}\in\mathbb{O}\subset\mathbb{R}^{m\times N}$,*

$$\partial\delta_{norm}\left(\boldsymbol{A}\right)=\left\{\boldsymbol{A}\tilde{\boldsymbol{L}},\tilde{\boldsymbol{L}}\in\mathbb{R}^{N\times N}:{\scriptstyle \boldsymbol{L}_{ii}\geq 0,\forall\, i \atop \tilde{\boldsymbol{L}}_{ij}=0, i\neq j}\right\}. \qquad (45)$$

*Proof.* Let $\boldsymbol{Z}\in\partial\delta_{\text{norm}}\left(\boldsymbol{A}\right)$. From the definition of the subgradient, we get $\delta_{\text{norm}}\left(\boldsymbol{A}\right)+\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\left(\boldsymbol{B}-\boldsymbol{A}\right)\right\}\leq\delta_{\text{norm}}\left(\boldsymbol{B}\right)$, $\forall\boldsymbol{B}\in\mathbb{R}^{m\times N}$. This relation is trivially satisfied for all $\boldsymbol{Z}$ and for any $\boldsymbol{B}\notin\mathbb{O}$. However, when $\boldsymbol{B}\in\mathbb{O}$, $\boldsymbol{Z}$ should satisfy

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}\leq\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}, \qquad (46)$$

since $\delta_{\text{norm}}\left(\boldsymbol{A}\right)=\delta_{\text{norm}}\left(\boldsymbol{B}\right)$.

To prove the result, we consider three different cases that cover all possible values for $\boldsymbol{Z}$.

1) We express the columns of the matrix $\boldsymbol{Z}$ as $\boldsymbol{Z}_i=\tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i+\boldsymbol{A}_i^{\perp}$, where $\tilde{\boldsymbol{L}}_{ii}\in\mathbb{R}$ and $\boldsymbol{A}_i^{\perp}\in\mathbb{R}^m$ is such that $\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i^{\perp}=0$, $\forall i$. Suppose $\boldsymbol{A}_i^{\perp}\neq\boldsymbol{0}$ for at least one value of $i$. Also, let $\boldsymbol{B}\in\mathbb{R}^{m\times N}\in\mathbb{O}$ be defined as

$$\boldsymbol{B}_i=\begin{cases}\boldsymbol{e}, & \text{for } \left\|\boldsymbol{Z}_i\right\|=0,\\ \boldsymbol{Z}_i/\left\|\boldsymbol{Z}_i\right\|, & \text{for } \left\|\boldsymbol{Z}_i\right\|\neq 0,\end{cases} \qquad (47)$$

where $\boldsymbol{e}$ is any unit norm vector. Then,

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}=\sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}<\sum_{i=1}^{N}\left\|\boldsymbol{Z}_i\right\|=\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}.$$

Therefore, there exists a matrix $\boldsymbol{B}\in\mathbb{O}$ such that (46) is not satisfied. Thus, we get

$$\partial\delta_{\text{norm}}\left(\boldsymbol{A}\right)\subseteq\left\{\boldsymbol{A}\tilde{\boldsymbol{L}},\tilde{\boldsymbol{L}}\in\mathbb{R}^{N\times N}:\tilde{\boldsymbol{L}}_{ij}=0,\ \text{if } i\neq j\right\}. \qquad (48)$$

2) Let $\boldsymbol{Z}=\boldsymbol{A}\tilde{\boldsymbol{L}}$ for some diagonal matrix such that at least one of the diagonal entries of $\tilde{\boldsymbol{L}}$ is negative. Let $\boldsymbol{B}\in\mathbb{R}^{m\times N}\in\mathbb{O}$ be defined such that $\boldsymbol{B}_i=\text{sign}\left\{\tilde{\boldsymbol{L}}_{ii}\right\}\boldsymbol{A}_i$, where the function sign$\{\cdot\}$ takes values 1 and $-1$ for nonnegative and negative arguments, respectively. Then,

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}=\sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}<\sum_{i=1}^{N}\left|\tilde{\boldsymbol{L}}_{ii}\right|\leq\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}, \qquad (49)$$

Therefore, (46) does not hold for $\boldsymbol{B}\in\mathbb{O}$, and from (48) we get

$$\partial\delta_{\text{norm}}\left(\boldsymbol{A}\right)\subseteq\left\{\boldsymbol{A}\tilde{\boldsymbol{L}},\tilde{\boldsymbol{L}}\in\mathbb{R}^{N\times N}:{\scriptstyle \boldsymbol{L}_{ii}\geq 0,\forall i \atop \tilde{\boldsymbol{L}}_{ij}=0,\ \text{if } i\neq j}\right\}. \qquad (50)$$

3) Let $\boldsymbol{Z}=\boldsymbol{A}\tilde{\boldsymbol{L}}$, for some psd matrix $\tilde{\boldsymbol{L}}$. Here, for any matrix $\boldsymbol{B}\in\mathbb{O}$,

$$\begin{aligned}
\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}=\mathsf{Tr}\left\{\tilde{\boldsymbol{L}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}\right\}&=\sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{B}_i \qquad (51)\\
&\leq\sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}=\sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i \qquad (52)\\
&=\mathsf{Tr}\left\{\tilde{\boldsymbol{L}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right\}=\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}. \qquad (53)
\end{aligned}$$

Therefore, from (50) we get

$$\partial\delta_{\text{norm}}\left(\boldsymbol{A}\right)=\left\{\boldsymbol{A}\tilde{\boldsymbol{L}},\tilde{\boldsymbol{L}}\in\mathbb{R}^{N\times N}:{\scriptstyle \boldsymbol{L}_{ii}\geq 0,\forall i \atop \tilde{\boldsymbol{L}}_{ij}=0, i\neq j.}\right\}. \qquad (54)$$

Hence, the proof is complete. □

### A. *Proof of Theorem 1*

*Proof.* In [41, Theorem 2], the authors provide a Kurdyka-Łojasiewicz property based proof of convergence of a proximal algorithm. By careful examination their proof, it can be seen that a bounded sequence of iterates converges to a stationary point of $\tilde{g}$ if the following four conditions hold:[7]

(i) The function $\tilde{g}(\boldsymbol{A})$ satisfies $\inf\limits_{\boldsymbol{A}\in\mathbb{R}^{m\times N}}\tilde{g}\left(\boldsymbol{A}\right)>-\infty$.

(ii) There exist constants $\theta\in[0,1)$, $C,\epsilon>0$ such that

$$\left|\tilde{g}\left(\boldsymbol{A}\right)-\tilde{g}\left(\boldsymbol{A}^{*}\right)\right|^{\theta}\leq C\left\|\boldsymbol{Z}\right\| \qquad (55)$$

for any stationary point $\boldsymbol{A}^*$ of $\tilde{g}$, any $\boldsymbol{A}$ such that $\left\|\boldsymbol{A}-\boldsymbol{A}^*\right\|\leq\epsilon$, and any $\boldsymbol{Z}$ such that $\boldsymbol{Z}\in\partial g\left(\boldsymbol{A}\right)$. The constant $\theta$ is called the *Łojasiewicz exponent.*

(iii) There exists $C_1>0$ such that

$$\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right)-\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)\geq C_1\left\|\boldsymbol{A}^{(r,u-1)}-\boldsymbol{A}^{(r,u)}\right\|^{2}. \qquad (56)$$

(iv) There exist $u_0>1$, $C_2>0$ and $\boldsymbol{Z}\in\partial g\left(\boldsymbol{A}^{(r,u)}\right)$ such that for all $u>u_0$

$$\left\|\boldsymbol{Z}\right\|\leq C_2\left\|\boldsymbol{A}^{(r,u-1)}-\boldsymbol{A}^{(r,u)}\right\|. \qquad (57)$$

Here, the first two conditions are on the cost function, and the last two are on the iterates. In [41, Theorem 2], these conditions are verified to hold for the proximal algorithm. The rest of the proof below is the verification of the four conditions for the ALS procedure.

As discussed in Appendix A (see (32)), the cost function $g$ is bounded from below. Therefore, $\tilde{g}$ is also bounded from below, and hence Condition (i) is satisfied.

[7] A more detailed version of the proof precisely connecting it to the result in [46] is given in [30, Appendix D.12].

Next, note that $\delta_{\text{norm}}(\cdot)$ is an indicator function of a semi-algebraic set, and $g$ is a real analytic function. Therefore, $\tilde{g}$ is a sum of real analytic and semi-algebraic functions. As a consequence, from [47, Section 2.2], we have that $\tilde{g}$ satisfies the desired condition ((ii)).

Condition (iii) follows from Lemma 1.

Finally, to verify Condition (iv), we first compute the subgradient of $\tilde{g}$ using Lemma 2. Hence, the desired condition is true if and only if, for all $u > u_0$, it holds that

$$\min_{\tilde{\boldsymbol{Z}}\in\partial\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)} \left\|\tilde{\boldsymbol{Z}}\right\| \le C_2 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|. \tag{58}$$

Now, from Lemma 2, we have,

$$\min_{\tilde{\boldsymbol{Z}}\in\partial\tilde{g}(\boldsymbol{A})} \left\|\tilde{\boldsymbol{Z}}\right\|^2 = \min_{\tilde{\boldsymbol{L}}_{ii}\ge 0} \left\|\nabla g(\boldsymbol{A}) + \boldsymbol{A}\tilde{\boldsymbol{L}}\right\|^2. \tag{59}$$

Since the optimization problem is separable in the diagonal entries of $\tilde{\boldsymbol{L}}$, we get the optimum value $\tilde{\boldsymbol{L}}^*$ as

$$\tilde{\boldsymbol{L}}_{ii}^* = \begin{cases} -\boldsymbol{A}_i^{\mathsf{T}}\nabla g\left(\boldsymbol{A}\right)_i, & \text{if } \boldsymbol{A}_i^{\mathsf{T}}\nabla g\left(\boldsymbol{A}\right)_i \le 0 \\ 0, & \text{otherwise} \end{cases} \tag{60}$$

for $i = 1, 2, \ldots, N$. This gives

$$\begin{aligned} &\arg\min_{\tilde{\boldsymbol{Z}}\in\partial\tilde{g}(\boldsymbol{A})} \left\|\tilde{\boldsymbol{Z}}\right\| \\ &\le \sqrt{\sum_{i=1}^{N} \max\left\{\left\|\left(\boldsymbol{I} - \boldsymbol{A}_i\boldsymbol{A}_i^{\mathsf{T}}\right)\nabla g\left(\boldsymbol{A}\right)_i\right\|, \left\|\nabla g\left(\boldsymbol{A}\right)_i\right\|\right\}} \\ &= \left\|\nabla g\left(\boldsymbol{A}\right)\right\|. \end{aligned} \tag{61}$$

Here, (61) follows from the fact that $\boldsymbol{I} - \boldsymbol{A}_i\boldsymbol{A}_i^{\mathsf{T}}$ is the projection matrix for the subspace orthogonal to the unit norm column $\boldsymbol{A}_i$. Therefore, $\left\|\left(\boldsymbol{I} - \boldsymbol{A}_i\boldsymbol{A}_i^{\mathsf{T}}\right)\nabla g\left(\boldsymbol{A}\right)_i\right\| \le \left\|\nabla g\left(\boldsymbol{A}\right)_i\right\|$. Thus, we have

$$\min_{\tilde{\boldsymbol{Z}}\in\partial\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)} \left\|\tilde{\boldsymbol{Z}}\right\| = \left\|\left(\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right)\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\| \tag{62}$$

$$\le C_2 \left\|\left(\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right)\right\|, \tag{63}$$

where $C_2$ is the spectral norm of $\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$. Also, (62) is due to the definition of $g$ in (8). Hence, Condition (iv) is satisfied for all $u$. Therefore, all four conditions are met, and consequently, the convergence is guaranteed. □

## Appendix D
## Proof of Proposition 3

*Proof.* From (13) and Proposition 1,

$$\boldsymbol{0} = \lim_{u\to\infty} \boldsymbol{Z}^{(r,u)} = P_{\boldsymbol{A}^{(r)}}\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)}\boldsymbol{\Sigma}\right). \tag{64}$$

Thus, (16) gives

$$\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right) = \boldsymbol{A}^{(r)}\boldsymbol{L}, \tag{65}$$

for some diagonal $\boldsymbol{L}$. Then, the result related to the Nash equilibrium follows from Corollary 1. Further, we have

$$\nabla g\left(\boldsymbol{A}^{(r)}\right) = -\boldsymbol{A}^{(r)}\boldsymbol{L}. \tag{66}$$

Let $\boldsymbol{\Delta} = \boldsymbol{A} - \boldsymbol{A}^{(r)}$, where $\boldsymbol{A}$ is any matrix in $\mathbb{O}$. Then, for $i = 1, 2, \ldots, N$ we have

$$1 = \left\|\boldsymbol{A}_i\right\|^2 = \left\|\boldsymbol{\Delta}_i + \boldsymbol{A}_i^{(r)}\right\|^2 = \left\|\boldsymbol{\Delta}_i\right\|^2 + 1 + 2\boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i^{(r)}. \tag{67}$$

Thus, we get $\frac{1}{2}\left\|\boldsymbol{\Delta}_i\right\|^2 = -\boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i^{(r)}$, and similarly, expanding $\left\|\boldsymbol{A}_i - \boldsymbol{\Delta}_i\right\|^2$, we get $\frac{1}{2}\left\|\boldsymbol{\Delta}_i\right\|^2 = \boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i$. Therefore,

$$\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}\right\} = -\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}^{(r)}\right\} = \frac{1}{2}\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\right\}. \tag{68}$$

Now, using a Taylor series expansion around $\boldsymbol{A}^{(r)}$, we have

$$\begin{aligned} &g\left(\boldsymbol{A}\right) - g\left(\boldsymbol{A}^{(r)}\right) \\ &= \mathsf{Tr}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\nabla g\left(\boldsymbol{A}^{(r)}\right) + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} && (69) \\ &= \mathsf{Tr}\left\{-\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}^{(r)}\boldsymbol{L} + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} && (70) \\ &= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{L} + \boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} && (71) \\ &= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Delta}\left(\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{\Delta}^{\mathsf{T}}\right\}, && (72) \end{aligned}$$

where we use (66) and (68) to get (70) and (71) respectively. Note that the Taylor series expansion is not an approximation here, as our cost function is quadratic. The right hand side of (72) is non-negative if and only if $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is positive semi-definite, and strictly positive if and only if $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is positive definite. Hence, the proof is complete. □

## Appendix E
## Proof of Theorem 2

*Proof.* The first part of the result directly follows from Proposition 1 and [31, Theorem 4.4.1] (See Theorem 7 in Appendix G.)

To prove the second part, suppose that $\boldsymbol{A}^{(r)}$ is a strict local minimum. Then, for any neighborhood $\mathcal{U}$ of $\boldsymbol{A}^{(r)}$, there exists $\epsilon > 0$ such that, in the closed ball $\mathcal{H}_\epsilon \subseteq \mathcal{U}$ around $\boldsymbol{A}^{(r)}$, $g(\boldsymbol{A}) > g(\boldsymbol{A}^{(r)})$ for all $\boldsymbol{A} \ne \boldsymbol{A}^{(r)} \in \mathcal{H}_\epsilon$. Here, the closed ball is defined as follows:

$$\mathcal{H}_\epsilon = \left\{\boldsymbol{A} \in \mathbb{O} : \left\|\boldsymbol{A} - \boldsymbol{A}^{(r)}\right\| \le \epsilon\right\}. \tag{73}$$

Moreover, from Lemma 1, we get

$$\begin{aligned} \left\|G(\boldsymbol{A}) - \boldsymbol{A}^{(r)}\right\| &\le \left\|G(\boldsymbol{A}) - \boldsymbol{A}\right\| + \left\|\boldsymbol{A} - \boldsymbol{A}^{(r)}\right\| && (74) \\ &\le C_1\left[g\left(G(\boldsymbol{A})\right) - g\left(\boldsymbol{A}\right)\right] + \left\|\boldsymbol{A} - \boldsymbol{A}^{(r)}\right\| && (75) \\ &\le C_1\left[g\left(\boldsymbol{A}\right) - g\left(\boldsymbol{A}^{(r)}\right)\right] + \left\|\boldsymbol{A} - \boldsymbol{A}^{(r)}\right\|, && (76) \end{aligned}$$

where the last step is because of Proposition 1 which gives $g(\boldsymbol{A}) \ge g\left(G(\boldsymbol{A})\right) \ge g(\boldsymbol{A}^{(r)})$. From Proposition 3, we know that $\boldsymbol{A}^{(r)}$ satisfies the relation:

$$\boldsymbol{A}^{(r)}\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right), \tag{77}$$

for some diagonal matrix $\boldsymbol{L}$. Following the same steps as (70)-(72), we get

$$0 < g\left(\boldsymbol{A}\right) - g\left(\boldsymbol{A}^{(r)}\right)$$

$$= \frac{1}{2}\mathsf{Tr}\left\{\left(\boldsymbol{A}-\boldsymbol{A}^{(r)}\right)\left(\boldsymbol{L}+\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\left(\boldsymbol{A}-\boldsymbol{A}^{(r)}\right)^{\mathsf{T}}\right\}$$

$$\le \frac{\lambda_{\max}}{2}\left\|\boldsymbol{A}-\boldsymbol{A}^{(r)}\right\|^{2}, \tag{78}$$

where $\lambda_{\max} > 0$ is the largest singular value of the matrix $(\boldsymbol{L}+\boldsymbol{\Sigma}-\mathcal{D}\{\boldsymbol{\Sigma}\})$. Thus, from (76),

$$\left\|G(\boldsymbol{A})-\boldsymbol{A}^{(r)}\right\| \le \frac{C_1\lambda_{\max}}{2}\left\|\boldsymbol{A}-\boldsymbol{A}^{(r)}\right\|^{2}+\left\|\boldsymbol{A}-\boldsymbol{A}^{(r)}\right\|. \tag{79}$$

Let $\epsilon' > 0$ be such that

$$\max_{\boldsymbol{A}\in\mathcal{H}_{\epsilon'}}\left\|G(\boldsymbol{A})-\boldsymbol{A}^{(r)}\right\| = \epsilon \le \left(\frac{C_1\lambda_{\max}}{2}\epsilon'+1\right)\epsilon'. \tag{80}$$

Therefore, for all $\boldsymbol{A}\in\mathcal{H}_{\epsilon'}$, $G(\boldsymbol{A})\in\mathcal{H}_{\epsilon}$. Now, using the proof technique used in [31, Theorem 4.4.2], we define the set

$$\mathcal{V} = \{\boldsymbol{A}\in\mathcal{H}_{\epsilon} : g(\boldsymbol{A}) < \alpha\} \subseteq \mathcal{H}_{\epsilon}, \tag{81}$$

where $\alpha = \min_{\boldsymbol{B}\in\mathcal{H}_{\epsilon}\backslash\mathcal{H}_{\epsilon'}} G(\boldsymbol{B})$ when $\epsilon' \le \epsilon$, and $\alpha = \infty$ otherwise. Note that, when $\epsilon' \le \epsilon$, $g(\boldsymbol{A}) \ge \alpha$, for all $\boldsymbol{A}\in\mathcal{H}_{\epsilon}\backslash\mathcal{H}_{\epsilon'}$, which implies $\mathcal{V}\subseteq\mathcal{H}_{\epsilon'}$. Also, when $\epsilon' > \epsilon$, $\mathcal{H}_{\epsilon'}\supset\mathcal{H}_{\epsilon'}\supseteq\mathcal{V}$. Thus, in both cases, $\mathcal{V}\subseteq\mathcal{H}_{\epsilon'}$. Hence, for every $\boldsymbol{A}\in\mathcal{V}$, $G(\boldsymbol{A})\in\mathcal{H}_{\epsilon}$. By Proposition 1, the sequence $g\left(G^{(u)}(\boldsymbol{A})\right)$ generated by ALS is nonincreasing, and hence

$$g\left(G(\boldsymbol{A})\right) \le g\left(\boldsymbol{A}\right) < \alpha. \tag{82}$$

Therefore, $G(\boldsymbol{A})\in\mathcal{V}$ for all $\boldsymbol{A}\in\mathcal{V}$, hence $G^{(u)}(\boldsymbol{A})\in\mathcal{V}\subseteq\mathcal{U}$ for all $u\in\mathbb{N}$. Thus, stability of the point is guaranteed. Since $\boldsymbol{A}^{(r)}$ is the only strict local minimum of $g$ in $\mathcal{V}$, it follows that $\lim_{u\to\infty}G^{(u)}(\boldsymbol{A}) = \boldsymbol{A}^{(r)}$ for all $\boldsymbol{A}\in\mathcal{V}$, which shows the asymptotic stability of $\boldsymbol{A}^{(r)}$. This completes the proof. □

## Appendix F
## Proof of Proposition 4

*Proof.* The AM and the ALS procedures along with the update equations of $\boldsymbol{\Gamma}$ ensure that

$$Q(\boldsymbol{\Lambda}^{(r)};\boldsymbol{\Lambda}^{(r-1)}) \le Q(\boldsymbol{\Lambda}^{(r-1)};\boldsymbol{\Lambda}^{(r-1)}), \forall r \ge 1. \tag{83}$$

This result immediately follows from Proposition 1 and the fact that (6) maximizes the part of $Q(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)})$ that depends on $\boldsymbol{\Gamma}_k$. Using the properties of EM [48], we have that

$$T(\boldsymbol{\Lambda}^{(r)}) \le T(\boldsymbol{\Lambda}^{(r-1)}). \tag{84}$$

Since $(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}})^{-1}$ is positive definite, from (3), we get

$$T(\boldsymbol{\Lambda}) \ge \sum_{k=1}^{K}\log\left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| \ge Km\log\sigma^2. \tag{85}$$

Therefore, $\left\{T(\boldsymbol{\Lambda}^{(r)})\right\}_{r\in\mathbb{N}}$ is monotonically decreasing and bounded from below. Hence, the sequence of DL-SBL cost function values converges. □

## Appendix G
## Proof of Theorem 3

Before we present the proof of the theorem, we first list a set of results from the literature that are used in the proof.

### A. Toolbox

**Definition 3** (Coercive function)**.** *A function $T:\mathbb{R}^N\to\mathbb{R}$ is called coercive if $\lim_{\|\boldsymbol{x}\|\to\infty}T(\boldsymbol{x}) = +\infty$.*

**Lemma 3.** *The cost function $T(\boldsymbol{\Lambda})$ defined in* (3) *is a coercive and continuous function of $\boldsymbol{\Lambda}$.*

*Proof.* The proof is adapted from the proofs of [25, Theorem 1, Corollary 1]. We have

$$\lim_{\|\boldsymbol{\Lambda}\|\to\infty}T(\boldsymbol{\Lambda}) = \lim_{\|\boldsymbol{\gamma}_k\|\to\infty}\sum_{k=1}^{K}\Big(\log\left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right|$$

$$+\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\Big) = \infty. \tag{86}$$

Therefore, $T(\boldsymbol{\Lambda})$ defined in (3) is a coercive function of $\boldsymbol{\Lambda}$, because $\|\boldsymbol{\Lambda}\|\to\infty$ only if at least one of the entries of $\{\boldsymbol{\gamma}_k\}_{k=1,2,\ldots,K}$ goes to $\infty$, and $\boldsymbol{A}$ belongs to a bounded set.

The continuity of the cost function with respect to $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}, k = 1,2,\ldots,K$ follows from the fact that both the determinant and matrix inverse functions are continuous [49, Theorems 5.18, 5.19]. Further, since $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}$ is a continuous function of $\boldsymbol{A}$ and $\boldsymbol{\gamma}_k, k = 1,2,\ldots,K$, the cost function is a continuous function of $\boldsymbol{\Lambda}$. □

**Theorem 6** ( [48, Theorem 1])**.** *Let $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ be the iterates generated by a generalized EM algorithm. Also, let $G$ be the point-to-set mapping defining algorithm updates: $\boldsymbol{\Lambda}^{(r+1)} = G(\boldsymbol{\Lambda}^{(r)})$. Then, all the limit points of $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ are the set of stationary points crit$(T)$ of the cost function $T$, if*

*(i) $T(\boldsymbol{\Lambda}^{(r)}) > T(\boldsymbol{\Lambda}^{(r-1)})$, for all $\boldsymbol{\Lambda}\notin$ crit$(T)$.*

*(ii) $G(\boldsymbol{\Lambda}^{(r-1)})$ is a closed set over the complement of crit$(T)$.*

**Theorem 7** ( [31, Theorem 4.4.1])**.** *For any set $\mathcal{G}$ Let $G:\mathcal{G}\to\mathcal{G}$ be a descent mapping for a smooth cost function $T:\mathcal{G}\to\mathbb{R}$, and assume that for every $\boldsymbol{\Lambda}\in\mathcal{G}$, all limit points of $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ are stationary points of $T$. Let $\hat{\boldsymbol{\Lambda}}\in\mathcal{G}$ be any limit point of $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ which is not a local minimum of $T$. Further, assume that there is a compact neighborhood $\mathcal{U}$ of $\hat{\boldsymbol{\Lambda}}$ where, for every critical point $\bar{\boldsymbol{\Lambda}}$ of $T$ in $\mathcal{U}$, $T(\bar{\boldsymbol{\Lambda}}) = T(\hat{\boldsymbol{\Lambda}})$. Then, $\hat{\boldsymbol{\Lambda}}$ is an unstable point of $T$.*

### B. Proof of Theorem 3

*Proof.* From Proposition 4, the sequence $\left\{T(\boldsymbol{\Lambda}^{(r)})\right\}_{r\in\mathbb{N}}$ generated by the DL-SBL algorithm using ALS procedure converges to a point $T(\boldsymbol{\Lambda}^{*})$. Also, from Lemma 3, we know that $T(\boldsymbol{\Lambda}^{*})$ is finite and therefore the set of limit points is compact. Thus, it follows that the iterates $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ admit at least one limit point for $k = 1,2,\ldots,K$.

Next, we use Theorem 6 to prove that the iterates converge to the set of stationary points of the cost function. Clearly, Condition (i) of Theorem 6 is satisfied due to Proposition 4 and the properties of E and M steps. To prove Condition (ii), we first note that the AM and the ALS algorithm converge to a closed set, as proved by Proposition 3. Further, since $T(\boldsymbol{\Lambda})$ is a continuous function of $\boldsymbol{A}$ and $\boldsymbol{\gamma}_k$, the M-step update always satisfies Condition (ii) of Theorem 6. Therefore, the algorithm satisfies both conditions, and hence, converges to the set of stationary points.

The last part of the result about the stability follows directly from Proposition 4 and Theorem 7. □

## Appendix H
## Proof of Proposition 5

*Proof.* Under noiseless measurements, the dictionary learning problem reduces to a matrix factorization problem: $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$. Suppose that $\boldsymbol{X}$ is already known to the algorithm. Then, to uniquely estimate $\boldsymbol{A}$, condition (24) is necessary. Similarly, when $\boldsymbol{A}$ is known to the algorithm, to uniquely recover $\boldsymbol{X}$, (25) is necessary. Otherwise, there exists an $s-$sparse vector $\boldsymbol{z}$ in the null space of $\boldsymbol{A}$ such that $\boldsymbol{z} + \boldsymbol{x}_k$ is $s-$sparse for some $k$, and $\boldsymbol{y}_k = \boldsymbol{A}(\boldsymbol{z} + \boldsymbol{x}_k)$, i.e., the solution is not unique. Also, for $\boldsymbol{X}$ to have full rank, at least two columns of $\boldsymbol{X}$ must have different supports. Therefore, if $|\mathcal{S}_k| = m$, uniqueness is not guaranteed, which leads to the condition $|\mathcal{S}_k| < m$. Thus, the first part of the result is obtained.

Next, if $\max_{k=1,2,\ldots,K} \|\boldsymbol{x}_k\|_0 = 1$, every nonzero measurement vector is a scaled version of some column of the measurement matrix. The condition (24) guarantees that there is no all-zero row in $\boldsymbol{X}$ and thus, there exists a measurement vector $\boldsymbol{y}_k$ corresponding to every column $\boldsymbol{A}_i$ of the dictionary such that $\boldsymbol{y}_k = \boldsymbol{X}_{ik}\boldsymbol{A}_i$ where $\boldsymbol{X}_{ik}$ is the only nonzero entry of the $k^{\mathsf{th}}$ column of $\boldsymbol{X}$. Further, by assumption, the columns of $\boldsymbol{A}$ are unit norm, and hence, given $\boldsymbol{y}_k$, the tuple $(\boldsymbol{X}_{ik}, \boldsymbol{A}_i)$ is unique upto the sign of $\boldsymbol{X}_{ik}$. Thus, the solution is unique under (24) and (25). □

## Appendix I
## Proof of Theorem 4

*Proof.* The cost function $T$ in (3) consists of two terms: the logarithm of the determinant of the product of matrices of the form $\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}$, and sum of projections of the inverses of the same matrices. Since the second term is positive, the minimum is achieved when the first term goes to minus infinity while maintaining a finite upper bound on the second term. We note that, from (25)

$$\mathsf{Rank}\left\{\boldsymbol{\Gamma}_k^*\right\} = \left\|\mathsf{Diag}\left\{\boldsymbol{\Gamma}_k^*\right\}\right\|_0 = |\mathcal{S}_k| < m, \tag{87}$$

where $\mathcal{S}_k$ denotes the support of $\boldsymbol{x}_k^*$. Further, we get that

$$\begin{aligned}&\lim_{\sigma^2\to 0}\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}\right| \\ &\quad\le \lim_{\sigma^2\to 0}(\hat{\lambda}_{max} + \sigma^2)^{\mathsf{Rank}\{\boldsymbol{\Gamma}_k^*\}}(\sigma^2)^{m-\mathsf{Rank}\{\boldsymbol{\Gamma}_k^*\}} = 0,\end{aligned} \tag{88}$$

where $\hat{\lambda}_{max}$ is the largest eigenvalue of $\boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}$. Thus, the first term goes to negative infinity. Using arguments similar to those in [29, Theorem 1], we can show that

$$\lim_{\sigma^2\to 0}\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}\right)^{-1}\boldsymbol{y}_k \le \frac{1}{c}\left\|\boldsymbol{x}_k^*\right\|^2. \tag{89}$$

Thus, the second term in the cost function is upper bounded by $\frac{1}{c}\left\|\boldsymbol{X}^*\right\|_F^2 < \infty$. Hence, $\left(\boldsymbol{A}^*, \{\boldsymbol{\Gamma}_k^*\}_{k=1}^K\right)$ achieves the global minimum. Further, it is easy to see that the cost function takes the same value over the set $\left(\boldsymbol{A}^*\boldsymbol{P}, \{\boldsymbol{P}\boldsymbol{\Gamma}_k^*\boldsymbol{P}\}_{k=1}^K\right)$, and thus the result is proved. □

## Appendix J
## Proof of Theorem 5

*Proof.* The goal of DL-SBL is to solve:

$$\begin{aligned}\min_{\boldsymbol{A}\in\mathbb{O}}\Bigg[\sum_{k=1}^K \min_{\boldsymbol{\gamma}_k\in\mathbb{R}_+^N}\log\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| \\ +\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\Bigg].\end{aligned} \tag{90}$$

For any given $\boldsymbol{A}$, the local minima of the objective function of the sub-optimization problem within the square brackets is at most $m-$sparse [29, Theorem 2]. Hence, the local minima of the DL-SBL cost function are all at most $m-$sparse. □

## References

[1] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 54, no. 12, pp. 3736–3745, Dec. 2006.

[3] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Nonlocal sparse models for image restoration," in *ICCV*, Sep. 2009, pp. 2272–2279.

[5] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariant sparse coding for audio classification," in *Proc. UAI*, Jul. 2007.

[6] M. Zibulevsky and B. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, Apr. 2001.

[7] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ICML*, Jun. 2007, pp. 759–766.

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[9] ——, "Supervised dictionary learning," in *Proc. Adv. in Neural Inform. Process. Syst.*, Dec. 2009, pp. 1033–1040.

[10] D. Bradley and J. Bagnell, "Differentiable sparse coding," in *Proc. Adv. in Neural Inform. Process. Syst.*, Dec. 2009, pp. 113–120.

[11] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proc. CVPR*, Jun. 2009, pp. 1605–1612.

[12] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. CVPR*, Jun. 2009, pp. 1794 – 1801.

[13] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[14] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proc. ICASSP*, Mar. 1999.

[15] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[16] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2178–2191, Jun. 2009.

[17] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (SimCO) for dictionary update and learning," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6340–6353, Dec. 2012.

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

[19] S. K. Sahoo and A. Makur, "Dictionary training for sparse representation as generalization of K-Means clustering," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 587–590, Jun. 2013.

[20] K. Schnass, "Convergence radius and sample complexity of ITKM algorithms for dictionary learning," *Appl. Comput. Harmo. A.*, vol. 45, no. 1, pp. 22–58, Jul. 2018.

[21] 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 Trans. Image Process.*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[22] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing - part II: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, Nov. 2014.

[23] L. Yang, J. Fang, H. Cheng, and H. Li, "Sparse Bayesian dictionary learning with a Gaussian hierarchical model," *Signal Process.*, vol. 130, pp. 93–104, Jan. 2017.

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

[25] I. Fedorov and B. D. Rao, "Multimodal sparse bayesian dictionary learning," *ArXiv e-prints*, May 2019. [Online]. Available: https://arxiv.org/abs/1804.03740

[26] I. Fedorov, B. D. Rao, and T. Q. Nguyen, "Multimodal sparse bayesian dictionary learning applied to multimodal data classification," in *Proc. ICASSP*, Mar. 2017, pp. 2237–2241.

[27] J. G. Serra, M. Testa, R. Molina, and A. K. Katsaggelos, "Bayesian k-svd using fast variational inference," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3344–3359, Mar. 2017.

[28] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–214, Sep. 2001.

[29] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[30] G. Joseph, "Linear dynamical systems with sparsity constraints: Theory and algorithms," Ph.D. dissertation, Indian Institute of Science, Bangalore, India, 2019.

[31] P.-A. Absil, M. Robert, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[32] P.-A. Absil and J. Malick, "Projection-like retractions on matrix manifolds," *SIAM J. Optim.*, vol. 22, no. 1, pp. 135–158, Oct. 2012.

[33] T. Kaneko, S. Fiori, and T. Tanaka, "Empirical arithmetic averaging over the compact Stiefel manifold," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 883–894, Feb. 2013.

[34] R. Hunger, "Floating point operations in matrix-vector calculus," Munich University of Technology, TUM-LNS-TR-05-05, Tech. Rep. TUM-LNS-TR-05-05, Sep. 2007.

[35] D. Wipf and S. Nagarajan, "Iterative reweighted $\ell_1$ and $\ell_2$ methods for finding sparse solutions," *IEEE J. Sel. Topics Sig. Proc.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[36] J. Cruz Neto, L. De Lima, and P. R. Oliveira, "Geodesic algorithms in Riemannian geometry," *Balkan J. Geom. Appl*, vol. 3, no. 2, pp. 89–100, 1998.

[37] J. Cruz Neto, O. Ferreira, and L. R. Lucambio Perez, "A proximal regularization of the steepest descent method in Riemannian manifold," *Balkan J. Geom. Appl*, vol. 4, no. 2, pp. 1–8, 1999.

[38] Y. Yang, "Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization," *J. Optim. Theory Appl.*, vol. 132, no. 2, pp. 245–265, 2007.

[39] C. Li and J. Wang, "Newton's method for sections on Riemannian manifolds: Generalized covariant $\alpha$-theory," *J. Complex.*, vol. 24, no. 3, pp. 423–451, Jun. 2008.

[40] X.-b. Li, N.-j. Huang, Q. H. Ansari, and J.-C. Yao, "Convergence rate of descent method with new inexact line-search on Riemannian manifolds," *J. Optim. Theory Appl.*, pp. 1–25, Sep. 2018.

[41] H. Attouch and J. Bolte, "On the convergence of the proximal algorithm for nonsmooth functions involving analytic features," *Mat. Programming*, vol. 116, no. 1-2, pp. 5–16, Jan. 2009.

[42] H. Liu, W. Wu, and A. Man-Cho So, "Quadratic optimization with orthogonality constraints: Explicit Łojasiewicz exponent and linear convergence of line-search methods," *ArXiv e-prints*, Oct. 2015. [Online]. Available: https://arxiv.org/abs/1510.01025

[43] B. Gao, X. Liu, X. Chen, and Y. xiang Yuan, "On the Łojasiewicz exponent of the quadratic sphere constrained optimization problem," *ArXiv e-prints*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.08781

[46] M. D. Asic and D. D. Adamovic, "Limit points of sequences in metric spaces," *The American Mathematical Monthly*, vol. 77, no. 6, pp. 613–616, Jun.-Jul. 1970.

[44] M. Razaviyayn, H.-W. Tseng, and Z.-Q. Luo, "Dictionary learning for sparse representation: Complexity and algorithms," in *Proc. ICASSP*, May 2014, pp. 5247–5251.

[45] M. E. Muller, "A note on a method for generating points uniformly on N-dimensional spheres," *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.

[47] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[48] C. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Stat.*, vol. 11, no. 1, pp. 95–103, Mar. 1983.

[49] J. R. Schott, *Matrix analysis for statistics*. John Wiley & Sons, 2016.

**Geethu Joseph** received the B. Tech. degree in Electronics and Communication Engineering from National Institute of Technology, Calicut, India, in 2011, and the M. E. degree in Signal Processing and the Ph.D. degree in Electrical Communication Engineering (ECE), from the Indian Institute of Science (IISc), Bangalore, in 2014 and 2019, respectively. She was awarded the Prof. I. S. N. Murthy medal in 2014 for being the best M. E. (signal processing) student in the Dept. of ECE, IISc. She is currently a post-doctoral fellow at the Department of Electrical Engineering and Computer Science, Syracuse University, NY. Her research interests include statistical signal processing, adaptive filter theory, sparse Bayesian learning, and compressive sensing.

**Chandra R. Murthy** received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology Madras, Chennai, India, in 1998, the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Purdue University, West Lafayette, IN and the University of California, San Diego, CA, in 2000 and 2006, respectively. From 2000 to 2002, he worked as an engineer for Qualcomm Inc., San Jose, USA, where he worked on WCDMA baseband transceiver design and 802.11b baseband receivers. From 2006 to 2007, he worked as a staff engineer at Beceem Communications Inc., Bangalore, India on advanced receiver architectures for the 802.16e Mobile WiMAX standard. Currently, he is working as a Professor in the department of Electrical Communication Engineering at the Indian Institute of Science, Bangalore, India.

His research interests are in the areas of energy harvesting communications, multiuser MIMO systems, and sparse signal recovery techniques applied to wireless communications. His paper won the best paper award in the Communications Track at NCC 2014 and a paper co-authored with his student won the student best paper award at the IEEE ICASSP 2018. He has 60+ journal papers and 90+ conference papers to his credit. He was an associate editor for the IEEE Signal Processing Letters during 2012-16. He is an elected member of the IEEE SPCOM Technical Committee for the years 2014-16, and has been re-elected for the 2017-19 term. He is a past Chair of the IEEE Signal Processing Society, Bangalore Chapter. He is currently serving as an associate editor for the IEEE Transactions on Signal Processing and IEEE Transactions on Information Theory, and as an editor for the IEEE Transactions on Communications.