Improving the Incoherence of a Learned Dictionary via Rank Shrinkage

Shashanka Ubaru[1], **Abd-Krim Seghouane**[2] **and Yousef Saad**[1]
[1] Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN, USA.
[2] Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, Victoria, Australia

Keywords: Dictionary learning, mutual coherence, reduced rank, nonnegative garrotte.

Abstract

This letter considers the problem of dictionary learning for sparse signal representation whose atoms have low mutual coherence. To learn such dictionaries, at each step we first updated the dictionary using the Method of Optimal Directions (MOD), and then apply a dictionary rank shrinkage step to decrease its mutual coherence. In the rank shrinkage step, we first compute a rank one decomposition of the column normalized least squares estimate of the dictionary obtained from the MOD step. We then shrink the rank of this learned dictionary by transforming the problem of reducing the rank to a nonnegative garrotte estimation problem, and solving it using a path-wise coordinate descent approach. We establish theoretical results which show that the rank shrinkage step included will reduce the coherence of the dictionary, which is further validated by experimental results. Numerical experiments illustrating the performance of the proposed algorithm in comparison to various other well known dictionary learning algorithms are also presented.

1 Introduction

In recent years, sparse signal approximations by means of redundant or overcomplete dictionaries have received a lot of attention across various research areas, particularly in signal and image processing (Elad, 2010; Rubinstein et al., 2010). Considering a set of signals $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ with $\mathbf{y}_i \in \mathbb{R}^n$ and a redundant (overcomplete) dictionary $\mathbf{D} \in \mathbb{R}^{n\times K}$, $K > n$, the sparse signal approximation model assumes that the signals $\mathbf{y}_i$ can be represented as a sparse linear combination of the columns of $\mathbf{D}$, which are also called atoms. So, we can express this model as

$$\mathbf{y}_i \simeq \mathbf{D}\mathbf{x}_i, \quad i = 1, \ldots, N, \tag{1}$$

where $\mathbf{x}_i$'s $\in \mathbb{R}^K$ are sparse vectors with a very small number of non-zero approximation coefficients $s = \| \mathbf{x}_i \|_0 \ll n$. The particularity of a dictionary learning model is that we learn the dictionary $\mathbf{D}$, as well as find the sparse linear multivariate model that best describes the set of signals $\mathbf{Y}$, simultaneously. The parameters of this model are determined by solving the sparse approximation problem

$$\underset{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X}}{\arg\min} \| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2, \tag{2}$$

where $\| \cdot \|_F$ is the Frobenius norm, and $\mathcal{D}$ and $\mathcal{X}$ are the admissible sets for the dictionary and the approximation coefficient matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, respectively. Th set $\mathcal{D}$ is usually defined as the set of all dictionaries with unit column norm, i.e., $\mathcal{D} = \{\mathbf{D} \in \mathbb{R}^{n\times K} : \|\mathbf{d}_k\|_2 = 1, \forall k\}$, whereas $\mathcal{X}$ constrains the coefficient matrix to be sparse, i.e., the number of nonzero entries s in $\mathbf{X}$ is much smaller compared to the total number of entries n or $\mathcal{X} = \{\mathbf{X} \in \mathbb{R}^{K\times N} : \| \mathbf{x}_i \|_0 \leq s \ll n, \forall i\}$. Dictionary learning starts with the set of signals $\mathbf{Y}$, and aims to find

both the dictionary $\mathbf{D}$ and the approximation coefficients matrix $\mathbf{X}$. The optimization problem (13) is not convex and has been shown to be an NP hard problem (Razaviyayn et al., 2015). So, the methods for solving it can only hope to achieve an approximate solution.

Several algorithms have been proposed in the literature to address the dictionary learning problem (Engan et al., 1999; Aharon et al., 2006; Elad, 2010; Mailhe et al., 2012; Barchiesi and Plumbley, 2013). Most of these algorithms are comprised of two stages: a *sparse coding stage* and a *dictionary update stage*. In the first stage, the dictionary $\mathbf{D}$ is fixed and the sparsity constraint is used to compute a sparse linear approximation $\mathbf{X}$ for the given signals $\mathbf{Y}$. In the second stage, using the current sparse approximation $\mathbf{X}$ (sometimes called codes), the dictionary $\mathbf{D}$ is updated such that a certain cost function is minimized. Different cost functions have been used in the literature for the dictionary update stage in order to achieve different objectives. For example, the Frobenius norm with column normalization has been widely used. The dictionary learning methods iterate between the sparse coding stage and the dictionary update stage until convergence. These algorithms differ in the details of the approaches used for estimating $\mathbf{X}$ and updating $\mathbf{D}$. Besides the differences in the approaches used to update the dictionary, the dictionary update approaches can be either sequential where each dictionary atom ($\mathbf{d}_k$, $k = 1, \ldots, K$, of $\mathbf{D}$) is updated separately, for example as in (Aharon et al., 2006; Sahoo and Makur, 2013; Seghouane and Hanif, 2015) or in parallel where the dictionary atoms are updated all at once as in (Engan et al., 1999; Kreutz-Delgado et al., 2003; Hanif and Seghouane, 2014) for example.

Not only do the dictionary learning algorithms aim at obtaining the best representation for the set of signals $\mathbf{Y}$, but they also aim at finding a dictionary $\mathbf{D}$ that results in the best *sparse* representation for $\mathbf{Y}$. In order to achieve this, the motivation is to generate a low coherence or incoherent dictionary, since such a dictionary will help in the sparse coding stage to find sparser representations or solutions, see Theorem 1 in (Cleju, 2014) and the arguments in (Tropp, 2004; Barchiesi and Plumbley, 2013) for details. It has also been shown that a low coherence results in a well conditioned dictionary (and sub-dictionaries) (Tropp, 2008). Many algorithms have been proposed for obtaining incoherent dictionaries, in the dictionary learning (Ramırez et al., 2009; Mailhe et al., 2012; Barchiesi and Plumbley, 2013) and compressed sensing (Cleju, 2014; Elad, 2007) literature. The coherence $\mu(\mathbf{D})$ is a property that characterizes the similarity between different atoms of the dictionary. It is defined as the maximum correlation of any two dictionary atoms

$$\mu(\mathbf{D}) = \max_{1 \leq i,j \leq N, i \neq j} | \langle \mathbf{d}_i, \mathbf{d}_j \rangle |, \tag{3}$$

also known as mutual coherence. An alternate measure that can be used to characterize the coherence property of dictionaries, and which is used in this paper is the average coherence (Bajwa et al., 2010) defined as

$$\nu(\mathbf{D}) = \max_{1 \leq j \leq N} \frac{1}{N-1} \sum_{i=1, i \neq j}^{N} | \langle \mathbf{d}_i, \mathbf{d}_j \rangle | . \tag{4}$$

In our experiments (section 5), we use the average coherence $\nu(\mathbf{D})$ to evaluate the performances of the various dictionary learning algorithms that are compared. A dictionary with small coherence is referred to as an incoherent dictionary.

In this letter we follow up on the MOD (Engan et al., 1999) algorithm and propose an improvement by reducing the coherence of the learned dictionary using a rank shrinkage step. This step is added to the dictionary update stage in order to learn a dictionary with reduced coherence that is adapted to the set of signals $\mathbf{Y}$. The rank shrinkage of the learned dictionary is obtained by transforming the problem of reducing the rank to a nonnegative garrotte estimation problem of a reshaped rank one decomposition of the dictionary, and solving this problem using a path-wise coordinate descent approach.

The rest of the paper is organized as follows: Section 2 provides a review of the MOD algorithm, as well as some of the approaches that are proposed in the literature for reducing the coherence of learned dictionaries. In section 3, the proposed approach for improved incoherent dictionary learning is presented and the step-by-step algorithm of the proposed approach is provided. In section 4, we establish theoretical results that connect the smallest nonzero singular value of the dictionary to its mutual coherence, and show how the rank shrinkage step reduces the coherence of the learned dictionary. Section 5 contains the numerical experimental results illustrating the performance of the proposed dictionary algorithm, in comparison with other popular dictionary learning algorithms. Concluding remarks are given in section 6.

2 Background

Finding the optimal solution to the problem (13) is difficult, if not impossible (has been shown to be an NP hard problem (Razaviyayn et al., 2015)). Splitting the problem into two stages, as described before, makes the problem more tractable. The iterative solution MOD (Method of Optimal Directions) proposed in (Engan et al., 1999), where a pursuit algorithm is used in the first stage of the iteration, results in a good but a suboptimal solution. Given the set of signals $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$, the first stage of MOD consists of estimating the sparse codes $\mathbf{x}_i$, $i = 1, \ldots, N$ that constitute the columns of the matrix $\mathbf{X}$ by fixing $\mathbf{D}$ and solving

$$\hat{\mathbf{x}}_i = \arg\min_{x_i} \parallel \mathbf{y}_i - \mathbf{D}\mathbf{x}_i \parallel_2; \textit{ subject to } \parallel \mathbf{x}_i \parallel_0 \leq s \quad i = 1, \ldots, N,$$

where $s \ll K$. Finding the optimal s corresponds to a model order selection problem that can be resolved using a univariate linear model section criterion (Seghouane, 2010). The vectors $\mathbf{x}_i$ are the sparse representations of the signals $\mathbf{y}_i$, and are computed using the current dictionary $\mathbf{D}$. Many sparse coding algorithms have been proposed to solve the above optimization problem, e.g. see (Chen et al., 2001; Pati et al., 1993; Tropp and Gilbert, 2007), that can be used in this stage. For additional details on the sparse coding stage, we refer (Aharon et al., 2006; Elad, 2010). In our algorithm and experiments, we use the Orthogonal Matching Pursuit (OMP) algorithm proposed in (Pati et al., 1993; Tropp and Gilbert, 2007).

The second stage of the dictionary learning iteration is the dictionary update stage, where $\mathbf{X}$ is fixed and $\mathbf{D}$ is computed. The MOD algorithm is perhaps the simplest algorithm that finds $\mathbf{D}$ by minimizing $\parallel \mathbf{Y} - \mathbf{D}\mathbf{X} \parallel_F^2$ with respect to $\mathbf{D}$. This amounts to

$$\mathbf{D} = \arg\min_{D} \parallel \mathbf{Y} - \mathbf{D}\mathbf{X} \parallel_F^2 = \mathbf{Y}\mathbf{X}^\top \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}. \tag{5}$$

In order to prevent $\mathbf{D}$ from being arbitrarily large and therefore have arbitrarily small values of $\mathbf{x}_i$, it is common to constrain $\mathbf{D}$ to belong to the set $\mathcal{D}$. This is achieved by normalizing the columns of $\mathbf{D}$ obtained by the least squares solution. However, the least square estimator (5) will at times perform poorly in both prediction and interpretation, if the generated atoms have high pairwise correlation, which will result in a poorly conditioned dictionary (Izenman, 1975). In order to characterize and monitor this pairwise correlation, the mutual coherence term (3) has been introduced. Hence, the aim of a dictionary learning algorithm is also to compute a dictionary $\mathbf{D}$ that is incoherent, i.e., $\mu(\mathbf{D})$ or $\nu(\mathbf{D})$ is small.

The low coherence of the dictionary can be enforced in the dictionary learning algorithm in two different ways. The first strategy is to add into the dictionary learning problem (13), a term that characterizes the incoherence objective of the learned dictionary. This lead to a modification of the dictionary update stage in order to promote mutually incoherent atoms. This is the approach adopted in (Ramirez et al., 2009) where the penalty term $\parallel \mathbf{G} - \mathbf{I} \parallel_F^2$, where $\mathbf{G} = \mathbf{D}^\top\mathbf{D}$ defines the Gram matrix, is used to enforce incoherence. The off-diagonal entries in $\mathbf{G}$ are the inner products that appear in (3). The mutual coherence $\mu(\mathbf{D})$ is therefore the largest off-diagonal entry of $\mathbf{G}$, and the average coherence $\nu(\mathbf{D})$ is the largest 1 norm of the columns of $(\mathbf{G} - \mathbf{I})$. The incoherence penalty used in (Ramirez et al., 2009) measure the Frobenius distance between the $\mathbf{G}$ and the identity matrix $\mathbf{I}$, which corresponds to the Gram matrix of an orthogonal dictionary whose mutual coherence is zero. This method is effective and is directly connected to the approximation accuracy (since the penalty term decides the error tolerated by the method). However, a drawback of this approach is that the coherence level of the resulting dictionary cannot be controlled directly.

An alternative strategy for learning incoherent dictionaries is to include a decorrelation step after the dictionary update stage to improve the incoherence of the dictionary at each iteration of the algorithm. This is the approach adopted in (Mailhe et al., 2012), where the incoherence of the resulting dictionary $\mathbf{D}$ from the dictionary update stage is improved by minimizing the Frobenius norm between $\mathbf{D}$ and the final dictionary $\mathbf{D}^*$, $\parallel \mathbf{D} - \mathbf{D}^* \parallel_F^2$ subject to the constraint $\mu(\mathbf{D}^*) \leq \mu_0$, where μ_0 is the targeted coherence. While this method allows us to set the coherence of the final dictionary to a desired value, the method is inconvenient since the incoherence is achieved independent of the input signals $\mathbf{Y}$. Hence, the method inevitably leads to a poorer approximation performance of the learned dictionary.

A similar strategy of including a decorrelation step was proposed in (Barchiesi and Plumbley, 2013), but with a different decorrelation method. Here, the decorrelation step is composed of three tasks. First, the off-diagonal entries of the Gram matrix $\mathbf{G} = \mathbf{D}^\top\mathbf{D}$ that are larger than the targeted coherence μ_0 are thresholded to the desired value $\mathbf{G} \leftarrow \text{Limit}(\mathbf{G}, \mu_0)$. Second, an eigen-decomposition of the resulting Gram matrix is computed $\mathbf{G} = \mathbf{Q}\Lambda\mathbf{Q}^\top$, and the first n eigenvector are used to reconstruct the dictionary $\bar{\mathbf{D}} = \Lambda^{1/2}\mathbf{Q}^\top$. In order to avoid the issue of $\bar{\mathbf{D}}$

being a poor approximation to the input signals $\mathbf{Y}$, this dictionary is further orthogonally rotated so as to obtain the best fit for $\mathbf{Y}$, i.e.; we minimize $\| \mathbf{Y} - \mathbf{W}\bar{\mathbf{D}}\mathbf{X} \|_F^2$. The final dictionary $\mathbf{D}$ at each iteration is obtained as $\mathbf{W}\bar{\mathbf{D}}$.

The proposed method in this paper uses the second strategy, i.e., invoke a decorrelation step after the dictionary update using MOD. However, we neither use a modified cost function for the dictionary update stage (as in (Ramirez et al., 2009)) nor directly control the coherence value of the dictionary (as in (Mailhe et al., 2012) and (Barchiesi and Plumbley, 2013)). Instead, we propose to use an additional step on the dictionary resulting from the dictionary update stage (5) (from MOD) that uses rank shrinkage to improve the incoherence at each iteration of the algorithm.

3 Incoherence via rank shrinkage

The multivariate model $\mathbf{Y} \simeq \mathbf{D}\mathbf{X}$ used in (13) typically does not account for the fact that the signals $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$ may be highly correlated, i.e., collinearity exists. In such situations, the initial dictionary chosen is likely to be poor, with near collinear atoms (columns). So, if multicollinearities exists, the mutual coherence of the dictionary will be high, since the correlation between the atoms will be high. As mentioned earlier, a highly correlated dictionary (with high coherence) will result is a poor sparse representation $\mathbf{X}$ (Tropp, 2004). This is because, most algorithms used in the sparse coding stage, e.g., orthogonal matching pursuit, rely on least squares solutions of the form $(\mathbf{D}^\top\mathbf{D})^{-\mathbf{1}}\mathbf{D}^\top\mathbf{Y}$ and then pick the largest coefficients to obtain a sparse solution. It is well known that a highly correlated matrix will result is a very poor least squares solution (due to poor condition number of the Gram matrix). Thus, it is desired to include a decorrelation step, i.e., have an incoherent dictionary at each stage of the updates. Tropp (Tropp, 2004) derived the incoherence condition required for the exact recovery of sparse signals using Orthogonal Matching Pursuit (OMP). He also showed that, he basis pursuit algorithms also require the same incoherence condition for exact recovery, see section 3.2 in (Tropp, 2004). The incoherent condition on the dictionary is also required for other iterative thresholding algorithms for sparse approximations, see (Schnass and Vandergheynst, 2008).

A popular method adopted in most applications to improve the performance of the least squares solution, is to reduce the rank of the highly correlated matrix (Izenman, 1975). The reduction of rank should identify and remove the multicollinearities within the atoms of $\mathbf{D}$ through its eigenvalues and eigenvectors. For the smaller (close to zero) eigenvalues λ_i of $\mathbf{G}$, and the associated eigenvector $\mathbf{v}_i$, we will have

$$\lambda_i = \mathbf{v}_i^\top \mathbf{D}^\top \mathbf{D}\mathbf{v}_i = (\mathbf{D}\mathbf{v}_i)^\top (\mathbf{D}\mathbf{v}_i) \simeq 0, \quad i = 1, \ldots, n-r$$

implying

$$\mathbf{D}\mathbf{v}_i \simeq 0, \quad i = 1, \ldots, n-r. \tag{6}$$

This implies that certain columns of the dictionary are near linearly dependent. Also note that, if the dictionary has near collinearity (near collinear atoms), these atoms (columns) of the dictionary are near linearly dependent. Removing the eigenpairs satisfying eq. (6), i.e., shrinking the rank, results in removing the contribution of these linearly dependent columns or near collinear atoms (we use a shrinkage operator that gives zero weights for these atoms). So, removing the eigenpairs close to zero will remove the collinearity within the atoms and improve the condition number of the dictionary (Gram matrix). Additional details on the relation between multicollinearity, correlation among atoms and the smaller eigenvalues of the Gram matrix are given in section 4. In section 4, we also provide theoretical justification that shows how our rank shrinkage step indeed reduces the coherence of the learned dictionary.

The proposed step to be included in the dictionary update stage is to improve the incoherence of the dictionary at each iteration of the algorithm is based on rank shrinkage. A simple approach for this would be to directly shrink the rank of $\mathbf{D}$ by shrinking (thresholding) the small eigenvalues of $\mathbf{G}$ to zero directly as done in (Barchiesi and Plumbley, 2013). However, this approach might not necessarily generate good results, since the shrinkage is performed independent of the input signals $\mathbf{Y}$. In order to overcome this issue, we present a new method for rank shrinkage based on the rank one decomposition of $\mathbf{D}$ obtained from (5), which we will call $\mathbf{D}_{OLS}$, and nonnegative garrotte estimation problem. The following constitutes the key contribution of this letter.

The reduced rank estimation or the rank shrinkage imposes a rank constraint on the dictionary $\mathbf{D}$, by estimating $\mathbf{D}$ under the constraint $\mathrm{rank}(\mathbf{D}) = r$ for $r \leq n$. First, we consider the rank one decomposition based on the singular value decomposition (SVD), and write the ordinary least square solution $\mathbf{D}_{OLS}$ in (5) as

$$\mathbf{D}_{OLS} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i=1}^{n} \hat{\mathbf{D}}_i, \tag{7}$$

where σ_i's are the singular values of $\mathbf{D}$, $\mathbf{u}_i$ and $\mathbf{v}_i$ are the corresponding left and right singular vectors, respectively, and $\hat{\mathbf{D}}_i = \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,\ i = 1, \ldots, n$ are rank one matrices. Hence, the estimate of $\mathbf{D}$ with rank $r \leq n$ that minimizes (13) (using the Eckart-Young theorem (Eckart and Young, 1936)) is given by

$$\mathbf{D}_r = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i=1}^{r} \hat{\mathbf{D}}_i. \tag{8}$$

Recall that the aim of reducing the rank of the predicted $\mathbf{D}_{OLS}$ is to reduce th multicollinearities within its atoms in the hope of improving the stability and prediction of the estimators ($\mathbf{D}$ and $\mathbf{X}$). This is equivalent to reducing the variance of $\mathbf{D}_{OLS}$ at the expense of the bias. When performed adequately, the reduction in variance may be large compared to the bias. So, this balance between the bias and the variance is controlled by r, which is the tuning parameter. However, since the rank shrinkage will be independent of the set of signals $\mathbf{Y}$, the dictionary $\mathbf{D}_r$ obtained need not be optimal. Hence, to obtain an improved reduced rank dictionary, we propose an adaptive shrinkage version of (8), based on the extension of (7) to

$$\mathbf{D}_{sh} = \sum_{i=1}^{n} \alpha_i \hat{\mathbf{D}}_i, \tag{9}$$

where α_i are weights assigned to the i^{th} component (rank one matrix) $\hat{\mathbf{D}}_i$. If $\alpha_i = 1$ for $i \leq r$ and 0 otherwise, we simply obtain back (8). Next, the idea is to control the weights of all the components using the information of signals $\mathbf{Y}$, and shrink some of the weights to zero, based on a criterion that involves $\mathbf{Y}$. This is achieved by considering the weights α_i as regression coefficients of a certain univariate linear model, and estimate them using a suitable sparsity promoting method. Since the weights α_i can not be negative, we propose to use the nonnegative garrotte (Breiman, 1995; Yuan and Lin, 2007) method as it is well suited for this situation.

In order to form the nonnegative garrotte, we first form a vector $\bar{\mathbf{y}}$ of length $(n * N)$ by vectorizing $\mathbf{Y}$. Similarly, we form n vectors $\bar{\mathbf{z}}_i$ of length $(n * N)$ by vectorizing $\hat{\mathbf{D}}_i \mathbf{X},\ i = 1, \ldots, n$. Next, we form a matrix $\mathbf{Z}$ of size$(n * N) \times n$ by concatenating $\bar{\mathbf{z}}_i$. Then, we estimate the sparse weight vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ using the nonnegative garrotte (Breiman, 1995) defined as

$$\begin{aligned} \boldsymbol{\alpha} &= \arg\min_{\boldsymbol{\alpha}} \| \bar{\mathbf{y}} - \mathbf{Z}\boldsymbol{\alpha} \|_2^2 + \lambda \sum_{i=1}^{n} \alpha_i \quad \text{subject to} \quad \alpha_i \geq 0 \\ &= \arg\min_{\boldsymbol{\alpha}} \| \bar{\mathbf{y}} - \sum_{i=1}^{n} \bar{\mathbf{z}}_i \alpha_i \|_2^2 + \lambda \sum_{i=1}^{n} \alpha_i \quad \text{subject to} \quad \alpha_i \geq 0. \end{aligned} \tag{10}$$

The optimization problem in (10) can be solved using a path-wise coordinate descent approach (Friedman et al., 2007). The coordinate-wise update for the weight vector $\boldsymbol{\alpha}$ will be of form

$$\alpha_i = \left(\frac{\bar{\mathbf{z}}_i^\top \bar{\mathbf{y}}_i - \lambda}{\bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_i} \right)_+,\ i = 1, \ldots, n \tag{11}$$

where $\bar{\mathbf{y}}_i = \bar{\mathbf{y}} - \sum_{j=1, j \neq i}^{n} \bar{\mathbf{z}}_i \alpha_i$. The parameter λ can be tuned to obtained the desired rank shrinkage in the dictionary $\mathbf{D}$, and in turn control its coherence. Thus, the proposed approach for the decorrelation of the learned dictionary is to use the above nonnegative garrotte rank shrinkage method.

Algorithm: The dictionary learning algorithm proposed in this paper is summarized in Algorithm 1, which includes the above described rank shrinkage step to obtain a more stable and incoherent dictionary $\mathbf{D}$.

4 Analysis

As described in the previous section, the purpose of the rank shrinkage step in Algorithm 1 is to remove collinearity between the atoms (decorrelate). In this section, we show that the rank shrinkage step indeed reduces the coherence of the dictionary.

Proposition 1 *The rank shrinkage step in Algorithm 1 reduces the mutual coherence $\mu(\mathbf{D})$ of the learned dictionary.*

Algorithm 1 Stepwise description of the proposed dictionary learning algorithm

Input: $\mathbf{Y}$, $\mathbf{D}_{ini}$, s, λ, ε and *J*.
Output: $\mathbf{D}, \mathbf{X}$.
Set $\mathbf{D} = \mathbf{D}_{ini}$.
For *it = 1 to J*
1. *Sparse Coding Stage:*
Find sparse coefficients $\mathbf{X}$, by approximately solving

$$\hat{\mathbf{x}}_i = \arg\min_{x_i} \| \mathbf{y}_i - \mathbf{D}\mathbf{x}_i \|^2; \textit{ subject to } \| \mathbf{x}_i \|_0 \leq s \quad i = 1, \ldots, N.$$

2. *Dictionary Update Stage:*
Generate the OLS solution $\mathbf{D}_{OLS} = \mathbf{Y}\mathbf{X}^\top \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}$.
2.a: Using the SVD compute $\mathbf{D}_{OLS} = \sum_{i=1}^{n} \hat{\mathbf{D}}_i$.
2.b: Construct the vectors $\bar{\mathbf{y}}$ and $\bar{\mathbf{z}}_i$, $i = 1, \ldots, n$.
2.c: Estimate the weights $\alpha_i,\ i = 1, \ldots, n$ as
While $||\boldsymbol{\alpha}^{iter} - \boldsymbol{\alpha}^{iter-1}||_2^2 \geq \varepsilon$ do
For $i = 1$ to n
Compute $\bar{\mathbf{y}}_i = \mathbf{y} - \sum_{j=1, j\neq i}^{n} \bar{\mathbf{z}}_i \alpha_i$
Compute the components $\alpha_i = \left(\frac{\bar{\mathbf{z}}_i^\top \bar{\mathbf{y}}_i - \lambda}{\bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_i}\right)_+$
end
$iter = iter + 1$
end while
2.d: Form $\mathbf{D} = \sum_{i=1}^{n} \alpha_i \hat{\mathbf{D}}_i$.
end

The following Lemma which gives a relation between the smallest non zero singular value and the mutual coherence of a matrix, and the arguments that follow this lemma provide the theoretical justification for the above proposition.

Lemma 1 *Given a dictionary* $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$*, such that* $\|\mathbf{d}_k\| = 1$*,* $k = 1, \ldots, K$ *and* $|\langle \mathbf{d}_i \mathbf{d}_j \rangle| < 1$ *for any* $i \neq j$*, Let us define* $\inf(\mathbf{D}) := \min_{\mathbf{t} \notin Null(\mathbf{D}), \|\mathbf{t}\|=1} \|\mathbf{D}\mathbf{t}\|$*. Suppose that*

$$\inf(\mathbf{D}) = \min_{\mathbf{t} \notin Null(\mathbf{D}), \|\mathbf{t}\|=1} \|\mathbf{D}\mathbf{t}\| \geq \sqrt{1-\eta}.$$

Then we have $\mu(\mathbf{D}) \leq \eta$*.*

Proof. For the given dictionary $\mathbf{D}$, let $\mathbf{G} = \mathbf{D}^\top \mathbf{D}$ define the corresponding Gram matrix. Let us write $\mathbf{G} = \mathbf{I} - \mathbf{H}$. The diagonal entries of $\mathbf{H}$ are zero. Then, for any vector $\mathbf{t}$ of norm 1, with $\mathbf{t} \notin Null(\mathbf{D})$, we have

$$\|\mathbf{D}\mathbf{t}\|^2 = \langle \mathbf{D}\mathbf{t}, \mathbf{t} \rangle = \langle (\mathbf{I} - \mathbf{H})\mathbf{t}, \mathbf{t} \rangle = 1 - \langle \mathbf{H}\mathbf{t}, \mathbf{t} \rangle \geq 1 - \eta.$$

As a result, for any vector $\mathbf{t} \notin Null(\mathbf{D})$:

$$\langle \mathbf{H}\mathbf{t}, \mathbf{t} \rangle \leq \eta. \tag{12}$$

Since the mutual coherence is the maximum off-diagonal entry of $\mathbf{G}$, we are interested in the off-diagonal entries $\mathbf{h}_{ij}$ (which are $-\mathbf{g}_{ij}$) for $i \neq j$. Let i, j be any pair with $i \neq j$ and take the vector

$$\mathbf{t}_{ij} = \frac{1}{\sqrt{2}}[\mathbf{e}_i + \sigma \mathbf{e}_j] \text{ with } \sigma = \text{sign}(\mathbf{h}_{ij}).$$

Then a little calculation shows that

$$\langle \mathbf{H}\mathbf{t}_{ij}, \mathbf{t}_{ij} \rangle = |\mathbf{h}_{ij}|.$$

Next, consider

$$\|\mathbf{D}\mathbf{t}_{ij}\|^2 = \frac{1}{2}\|\mathbf{d}_i \pm \mathbf{d}_j\|^2 = \frac{1}{2}[\|\mathbf{d}_i\|^2 + \|\mathbf{d}_j\|^2 \pm 2\langle \mathbf{d}_i, \mathbf{d}_j \rangle] = 1 \pm \langle \mathbf{d}_i, \mathbf{d}_j \rangle > 0.$$

This is due to the assumption that $|\langle \mathbf{d}_i, \mathbf{d}_j \rangle| < 1$ for $i \neq j$. Therefore, $\mathbf{t}_{ij}$ does not belong to the null space of $\mathbf{D}$, and from (12) we get $|\mathbf{h}_{ij}| < \eta$. This proves the lemma. □

Remark 1 *If we assume that* $\min_{\mathbf{t}\notin Null(\mathbf{D}),\|\mathbf{t}\|=1} \|\mathbf{D}\mathbf{t}\| \geq 1-\eta$*, then we would get a bound of the form* $\mu(\mathbf{D}) \leq \eta(2-\eta)$.

Observe that in Lemma 1, the term $\inf(\mathbf{D})$ is the smallest non-zero singular value of the dictionary $\mathbf{D}$. The argument is as follows: If $\inf(\mathbf{D})$ increases after rank shrinkage, one can choose a lower value for the parameter η in Lemma 1 such that $\inf(\mathbf{D}) \geq \sqrt{1-\eta}$, still holds. Then, by the statement of Lemma 1, $\mu(\mathbf{D}) \leq \eta$ satisfies for this lower value of η, which means $\mu(\mathbf{D})$ must reduce. Therefore, we need to show that the smallest non-zero singular value of the dictionary increases after our rank shrinkage step, for the coherence of the dictionary to decrease.

Let $\sigma_i, i = 1, \ldots, n$ be the singular values of the dictionary $\mathbf{D}$ before rank shrinkage labeled decreasingly, then $\inf(\mathbf{D}) = \sigma_n$. The singular values of the dictionary $\mathbf{D}_r$ after rank shrinkage will be $\alpha_i\sigma_i, i = 1, \ldots, r$. Thus, for Proposition 1 to hold, we need

$$\alpha_r\sigma_r > \sigma_n.$$

Clearly, $\sigma_r > \sigma_n$, and the coefficients α_i's are chosen such that they penalize only the smaller singular values (corresponding to near collinear subspaces). For the relevant atoms in the dictionary, the inner product $\bar{\mathbf{z}}_i^\top \bar{\mathbf{y}}_i$ in the updates of the coefficients α_i (eq. (11)), the coefficients which control the rank) will be high, and α_i will be large. So, such atoms are not removed (instead are promoted). Hence, we suggest to use a small tuning parameter λ in eq. (11). More importantly, since *the overall energy of the dictionary needs to be constant before and after rank shrinkage* since the columns of the dictionaries must have unit norms, the nonzero singular values of the rank shrunk matrix must be relatively higher to balance the energy from the singular values that were removed. This justifies Proposition 1.

A reduction in $\mu(\mathbf{D})$ will likely reduce the average coherence $\nu(\mathbf{D})$. We have the following relation between the two measures : $\nu(\mathbf{D}) \leq \frac{K}{K-1}\mu(\mathbf{D})$. (If the maximum value of correlation is reduced, the maximum of the average of correlations is likely to reduce.) In section 5, we demonstrate via many experiments that the rank shrinkage step proposed reduces both $\mu(\mathbf{D})$ and $\nu(\mathbf{D})$.

Coherence and the eigenvalues of the Gram matrix: In section 3, we saw how the eigenvalues of the Gram matrix $\mathbf{D}^\top\mathbf{D}$ that are close to zero (due to near collinearity) results in poor least squares solutions. The following argument gives further insight on the relation between the eigenvalues of the Gram matrix (which are squares of the singular values of the dictionary), collinearity, least squares solutions and mutual coherence.

Consider the sparse coding stage (updating $\mathbf{X}$). The columns of $\mathbf{X}$ in the sparse coding stage are obtained as the solutions of

$$\hat{\mathbf{x}}_i = \arg\min_x \|\mathbf{y}_i - \mathbf{D}_{Ii}\mathbf{x}\|_2^2 \qquad i = 1, ..., N,$$

and their variance-covariance matrix is given by

$$\text{Var}(\hat{\mathbf{x}}_i) = \sigma^2 \left(\mathbf{D}_{Ii}^\top\mathbf{D}_{Ii}\right)^{-1} = \sigma^2\mathbf{U}_{Ii}\Lambda_{Ii}^{-1}\mathbf{U}_{Ii} \qquad i = 1, ..., N, \tag{13}$$

where Ii represents the set of selected atoms with K_i cardinality, σ^2 is the noise variance and $\mathbf{D}^\top\mathbf{D}$ is the Gram matrix with eigen-decomposition $\mathbf{D}^\top\mathbf{D} = \mathbf{U}\Lambda\mathbf{U}^\top$ and rank r. We will drop the index i below.

The variance of each component of $\hat{\mathbf{x}}_i$ is given by

$$\text{Var}(\hat{x}_{i_j}) = \sigma^2 \sum_{l=1}^{r} \frac{u_{jl}^2}{\lambda_l} = \sigma^2 \sum_{l=1}^{r} \frac{a_{jl}^2}{\lambda_l^2},$$

where u_{jl} is the l^{th} component of $\mathbf{u}_j$ (column of $\mathbf{U}$) and $a_{jl} = u_{jl}\lambda_l$ is the l^{th} component of $\mathbf{a}_j = \mathbf{u}_j\Lambda$. Then, the total variance is given by

$$\sum_{j=1}^{r} \text{Var}(\hat{x}_{i_j}) = \sigma^2 \sum_{j=1}^{r}\sum_{l=1}^{r} \frac{a_{jl}^2}{\lambda_l^2} = \sigma^2 \sum_{l=1}^{r}\sum_{j=1}^{r} \frac{a_{jl}^2}{\lambda_l^2} = \sigma^2 \sum_{l=1}^{r} \frac{1}{\lambda_l}. \tag{14}$$

Now we observe that, if any one of the eigenvalues is very close to zero, the mean of the variances of the estimated sparse codes will increase to a very large extent. We saw in section 3 that, the eigenvalues are close to zero if the correlations between the dictionary atoms are high, that is, it depends on the extent of multicollinearity.

If the eigenvalues are decreasingly ordered $\lambda_1 \geq \ldots \geq \lambda_r$, then we have the following relations:

$$\sum_{l=1}^{r} \frac{1}{\lambda_l} \leq \sum_{l=1}^{r} \frac{1}{\lambda_r} = \frac{r}{\lambda_r}$$

and

$$\text{Var}(\hat{x}_{i_j}) = \sigma^2 \left(\mathbf{D}_{Ii}^\top \mathbf{D}_{Ii}\right)^{-1}_{jj} \geq \sigma^2,$$

which lead to

$$r\sigma^2 \leq \sum_{j=1}^{r} \text{Var}(\hat{x}_{i_j}) \leq \frac{r\sigma^2}{\lambda_1}.$$

If all the dictionary atoms are orthogonal (incoherent), the above inequality becomes an equality, since in this case, all eigenvalues are 1. If any one of the eigenvalues is close to zero, the variances of the estimated sparse codes will be very large according to (14). Hence, if the correlation among the dictionary atoms is considerable (coherent), the variance of the sparse code coefficients will be too large due to the inversion formula given in (13) (the analysis of the individual effect of the variables will be less meaningful here). Hence, it is necessary to quantify multicollinearity.

Incoherence and the mutual coherence $\mu(\mathbf{D})$ measure characterize the departure from orthogonality. If the dictionary atoms are strongly coherent, the matrix $\mathbf{D}_{Ii}^\top \mathbf{D}_{Ii}$ is ill-conditioned, and while the least squares estimator still exists, it will be very unstable (due to large variance). Recall that, the smallest eigenvalue of $\mathbf{D}_{Ii}$ is smaller than that of $\mathbf{D}$ by the interlacing theorem. Therefore, reducing the rank , i.e., a smaller r, will mean a larger nonzero smallest eigenvalue λ_r (in turn larger $\inf(\mathbf{D})$) and a reduced variance, and will result in more stable least squares solution.

In the next section, we present some experimental results which show that the smallest non-zero singular value of the dictionary after rank shrinkage will be larger than the smallest non-zero singular value before rank shrinkage. We will also illustrate via experiments that the coherence of the dictionary reduces after rank shrinkage.

5 Numerical Experiments

In this section, we illustrate the performance of the proposed dictionary learning scheme via several experiments, and compare its performance against other popular dictionary learning algorithms developed in the literature.

In the first experiment, we will demonstrate that the rank shrinkage step introduced in the dictionary update stage indeed reduces the overall coherence of the dictionary atoms obtained. For this, we first demonstrate that $\inf(\mathbf{D})$ (the smallest non-zero singular value) after rank shrinkage will be larger than $\inf(\mathbf{D})$ before rank shrinkage. Next, we will visualize the histogram of the Gram matrix $\mathbf{G}$ and average coherence $\nu(\mathbf{D})$ before and after rank shrinkage, since these give a better picture of the decorrelation than mutual coherence alone (which can be easily deduced from the histogram). As an illustration, we consider different types of dictionaries such as a random dictionary with Gaussian entries, dictionary formed using patches of images, and a dictionary with sine elements. We observe the behavior of their singular values (in Figure 1) and the coherence (in Figure 2) before and after the rank shrinkage step.

The random dictionary we considered is of size 20×50, and we generated $N = 1000$ signals $\mathbf{Y}$ from this dictionary with $s = 4$. First, we run Algorithm 1 on this $\mathbf{Y}$ for 20 iterations with the tuning parameter $\lambda = 10$. Figure 1(A) gives the singular value distributions of the dictionary $\mathbf{D}$ before and after rank shrinkage (after 10 iterations). Figure 1(B) gives the singular value distributions of the dictionary $\mathbf{D}$, when the algorithm was run with $\lambda = 30$. We observe that $\inf(\mathbf{D})$ after rank shrinkage is larger than $\inf(\mathbf{D})$ before rank shrinkage in both cases. We also observe how the value of the tuning parameter λ affects the amount of rank shrinkage achieved. Next, we observe the coherence of the dictionary before and after the rank shrinkage step. The top right plot in figure 2 shows the average coherence of the dictionaries obtained before (blue circles) and after (red stars) rank shrinkage over the 20 iterations for $\lambda = 10$. The top left plot gives the histogram of the entries of the Gram matrices $\mathbf{G}$ before and after rank shrinkage in the first iteration. Clearly, we observe that the coherence of the dictionary has reduced after the rank shrinkage step.

The atoms of the second dictionary considered were constituted of 8×8 image patches vectorized, and $K = 100$ of these atoms were used as the dictionary (size of the dictionary was 64×100). Algorithm 1 was run with with the tuning parameter $\lambda = 5$. Figure 1(C) gives the singular value distributions of the dictionary $\mathbf{D}$ before and after rank shrinkage (after 10 iterations). We observe that for such image dictionaries learned by MOD, many of

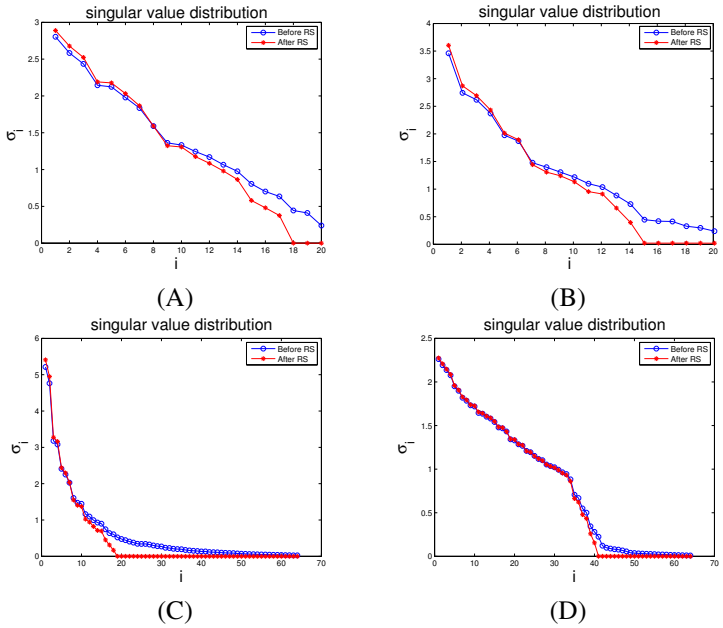

Figure 1: Singular value distributions before and after rank shrinkage. (A) For random dictionary with paarameter $\lambda = 10$ and (B) with $\lambda = 30$. (C) For image dictionary and (D) for a dictionary with sine elements.

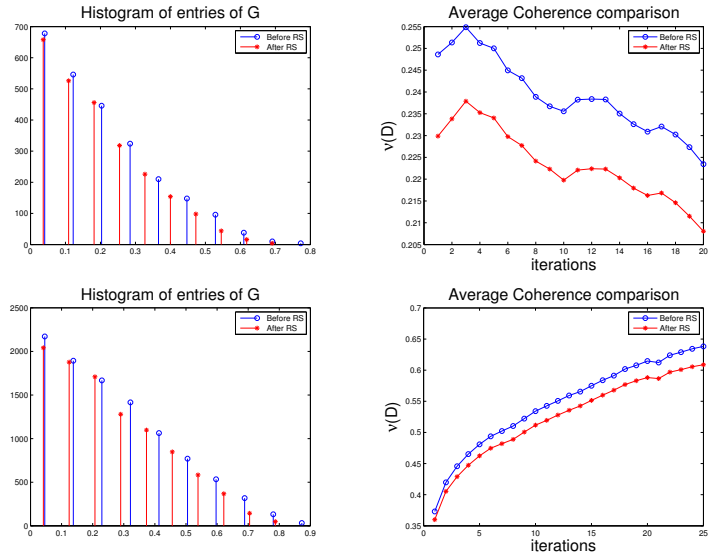

Figure 2: Histogram of the Gram matrix and the average coherence at different iterations before and after rank shrinkage for (top) random dictionary and (bottom) image dictionary.

the singular values are very close to zero indicating high multicollinearity amongst the dictionary atoms. It is a good idea to remove these collinearity between atoms to obtain a better dictionary, which we achieve using the rank shrinkage step proposed. To observe the coherence, we ran the algorithm 1 for 25 iterations and the average coherence of the dictionaries obtained before and after rank shrinkage were plotted in the bottom right plot of figure 2. The histogram of Gram matrices are plotted in the bottom left plot. Clearly, we observe that the coherence of the dictionary reduced after the rank shrinkage step.

To further illustrate how multicollinearity exists in the dictionaries, we consider a third dictionary of size 64×100 with atoms composed of sine elements and corners. Figure 1(D) gives the singular value distributions of this dictionary before and after rank shrinkage. We again observe that many singular values of the dictionary learned by MOD are very close to zero and our algorithm removes these near collinear subspaces and reduces the coherence of the learned dictionary.

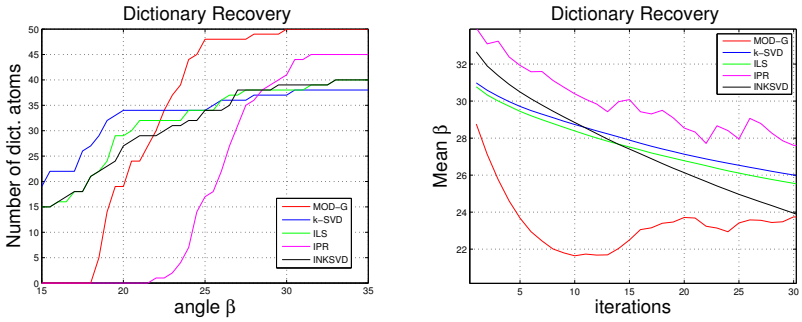

Figure 3: Recovery of a random dictionary with uniform distribution. (Left) Number of dictionary atoms recovered that are within β angle from the actual atoms. (Right) Average angle β after each iteration.

In the following experiments, we illustrate the performance of the proposed algorithm on different dictionary learning applications, and compare its performance against other popular dictionary learning algorithms, namely ILS (Iterative Least Squares) (Engan et al., 1999), K-SVD[1] (Aharon et al., 2006), IPR (Iterative Projections and Rotations) (Barchiesi and Plumbley, 2013) and INK-SVD[2] (Mailhe et al., 2012).

Dictionary Recovery: In this experiment, we consider the dictionary recovery problem of a small random dictionary with uniform distribution. We generate $N = 2000$ signals $\mathbf{Y}$ from a random dictionary with uniform distribution $\mathbf{D}$ with $n = 20, K = 50$ and $s = 4$. We add a small amount of noise to the signals $\mathbf{Y}$ (the strength of the signal is $SNR = 20dB$). The objective is to recover a dictionary $\tilde{\mathbf{D}}$ from the signals $\mathbf{Y}$ that is as close to the original dictionary $\mathbf{D}$ as possible using the dictionary learning (DL) algorithms. Figure 3 compares the results obtained for the five different dictionary learning (DL) algorithms considered.

The left plot in figure 3 plots the number of dictionary atoms (averaged over 10 trials) in the dictionary recovered that are within the given angle β (in degrees) from the original dictionary atoms for the five different dictionary learning algorithms. The angle β is defined as

$$\beta_i = \frac{\cos^{-1}(\langle \tilde{d}_i, d_i \rangle)}{\|\tilde{d}_i\|_2 \|d_i\|_2}, \; i = 1, \ldots, K.$$

All algorithms were run for 50 iterations and 10 trials. For the proposed algorithm (MOD-G, for MOD+garotte) we set $\lambda = 10$, and for INKSVD and IPR we set $\mu_0 = 0.8$ (the parameters were tuned until we obtain similar average coherence by all the three methods). In all the five algorithms, for the sparse coding stage, we use the Orthogonal Matching Pursuit (OMP) algorithm (Pati et al., 1993; Tropp and Gilbert, 2007), and we initialize the dictionary $\mathbf{D}_{ini}$ with randomly chosen samples of $\mathbf{Y}$. We observe that the dictionary atoms recovered by the proposed algorithm are better (more atoms that are closer to the original) than the other dictionary learning algorithms. The right plot in figure 3 plots the mean angle β of the dictionaries obtained at each iteration for the different DL algorithms. We observe that the proposed algorithm requires fewer iterations (converges faster) to recover a closer dictionary than the other algorithms.

Signal representation: In the following experiments, we compare the performance of the proposed algorithm against the other incoherent dictionary learning algorithms, namely IPR and INKSVD. The other two methods ILS and KSVD do not guarantee incoherent dictionaries and we cannot tune parameters to obtain a desired coherence.

In the next experiment, the objective is to obtain a sparse representation for a given set of signals using the dictionary learning algorithms. The underlining dictionary of the signals is unknown and the goal is to try and represent the signals $\mathbf{Y}$ using a small sized dictionary $\tilde{\mathbf{D}}$ and a sparse set of coefficients $\tilde{\mathbf{X}}$ obtained from the DL algorithms, such that the SNR of the signal representation $\tilde{\mathbf{Y}} = \tilde{\mathbf{D}}\tilde{\mathbf{X}}$ is high as possible. Figure 4 shows the performance of the three incoherent dictionary learning algorithms in obtaining a sparse representation for a set of $N = 2000$ signals $\mathbf{Y}$ generated from an autoregressive model of order 1, i.e., AR(1) signals.

[1] Codes for ILS and KSVD, and OMP for sparse coding were obtained from `http://www.ux.uis.no/~karlsk/dle/`.

[2] Codes for IPR and INKSVD were obtained from the SmallBox software `http://www.small-project.eu/software-data/smallbox/` (Damnjanovic et al., 2010).

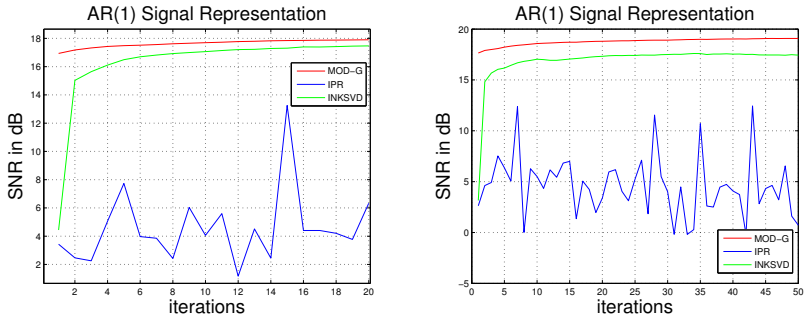

Figure 4: Sparse representation of AR(1) signals. (Left) Number of sparse coefficients $s = 4$ and number of iterations $J = 20$. (Right) $s = 5$ and $J = 50$.

Table 1: Dictionary Learning with of various parameters

Dictionary (Parameters)	Rank Shrinkage				IPR		INKSVD	
	λ	SNR	$\nu(D)$	μ_0	SNR	$\nu(D)$	SNR	$\nu(D)$
Uniform Random, $n = 20$, $K = 50$, $N = 2000$	2	13.36	0.79	0.9	12.3	0.83	13.17	0.88
	5	13.31	0.71	0.8	12.1	0.70	12.96	0.72
	10	13.0	0.63	0.7	12.4	0.60	13.05	0.63
	20	12.74	0.57	0.6	12.4	0.52	12.81	0.53
AR(1) model, $n = 16$, $K = 80$, $N = 2000$	10	18.5	0.67	0.9	16.2	0.67	18.01	0.65
	20	18.3	0.59	0.8	14.8	0.60	17.8	0.57
	50	17.9	0.51	0.7	11.9	0.55	17.6	0.53
	100	17.2	0.43	0.6	11.3	0.49	17.3	0.45
Image patches, $n = 64$, $K = 100$, $N = 2000$	5	12.5	0.64	0.9	11.1	0.78	12.3	0.73
	10	12.3	0.57	0.8	10.7	0.68	11.7	0.60
	20	11.2	0.51	0.7	10.5	0.59	11.3	0.50
	30	10.8	0.42	0.6	9.5	0.47	11.0	0.46

The two plots in figure 4 plot the signal to noise ratio (SNR$= 10\log_{10}(\|\tilde{\mathbf{Y}}\|_F/\|\mathbf{Y} - \tilde{\mathbf{D}}\tilde{\mathbf{X}}\|_F)$ of the signal representations $\tilde{\mathbf{Y}}$ obtained from the DL algorithms at each iterations. In the left plot, we considered the number of dictionary atoms to be $K = 50$, $s = 4$ and the number of iterations $J = 20$. In the right plot, we chose $K = 100, s = 5$ and $J = 50$. In both cases, we chose $\lambda = 18$ for MOD-G and $\mu_0 = 0.7$ for IPR and INKSVD (tuned to get similar average coherence ν of around 0.57 in the first example after 20 iterations). We observe that, the performance of the proposed method (MOD-G) is comparable and (slightly) better than INKSVD, and IPR performs poorly for this signal representation experiment.

Table 1 lists the signal to noise ratio (SNR$= 10\log_{10}(\|\mathbf{Y}\|_F/\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F)$ and the average coherence $\nu(\mathbf{D})$ for the dictionaries $\mathbf{D}$ and the coefficients $\mathbf{X}$ obtained from the three incoherent DL algorithms for different types of dictionaries, and with different parameters. For MOD-G, the shrinkage parameter λ was varied to control the coherence, while for IPR and INKSVD, we tuned μ_0 (but the same μ_0 for both) in order to control the coherence of the dictionaries obtained. All algorithms were run for $J = 100$ iterations. Clearly, we observe that the performance of the proposed method (MOD-G) is comparable and many times better than INKSVD, and always better than IPR. Although the SNRs obtained from INKSVD were comparable to our method, the dictionaries obtained by INKSVD were not close (in terms of the angle β) to the original dictionaries compared to the proposed method (MOD-G), as we saw in the dictionary recovery experiments. Also, we noted that our rank shrinkage method converges faster (takes few iterations to obtain closer dictionaries) compared to other methods.

Audio signal representation In the final experiment, we assess the performance of the proposed DL algorithm in recovering a set of audio signals, such that the dictionary has bounded average coherence and provides good approximation to the signals. For this, we will consider the same dataset considered in the INKSVD paper (Mailhe et al., 2012) and the IPR paper (Barchiesi and Plumbley, 2013), and compare the performances of the

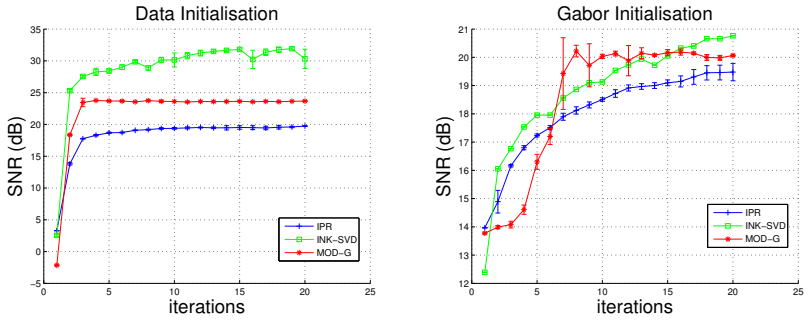

Figure 5: Audio signal representations. (Left) Data initialization. (Right) Gabor initialization.

three incoherent DL algorithms. The audio signals considered are excerpts of a $16kHz$ guitar recording named `music03_16kHz`, and is part of the data available in the SmallBox[3] toolbox (Damnjanovic et al., 2010). The toolbox also contains all the codes to reproduce the following experiment (expect the code for the proposed algorithm).

We consider exactly the same experiment demonstrated in (Mailhe et al., 2012; Barchiesi and Plumbley, 2013), where the recording is divided into 50% overlapping blocks of 256 samples (16 ms each) and the resulting signals is arranged as columns of the signal matrix $\mathbf{Y}$. The size of $\mathbf{Y}$ is 256×624. Next, the dictionary is initialized to be twice overcomplete, ie., the size of the dictionary is 256×512. The three incoherent DL algorithms (MOD-G, IPR and INKSVD) were run for 20 iterations, setting $s = 12$ non-zero coefficients in each sparse representation. In IPR and INKSVD, we set $\mu_0 = 0.5$ (for values below this, the performances of all three methods were poor and were hard to compare), and in MOD-G we set $\lambda = 16$, such that all three methods give $\nu(\mathbf{D})$ around 0.37 after 20 iterations. Figure 5 depicts the performances of the three methods when the dictionaries were initialized using i) a randomly chosen subset of the training data and ii) Gabor dictionary (Rubinstein et al., 2010).

The left plot of figure 5 plots the SNR of the recovered signal obtained by the three incoherent DL algorithms for each of iterations, when the dictionary was initialized using a randomly chosen subset of the training data. The right plot gives the SNR v/s iterations plot for the three algorithms for Gabor dictionary initialization. The experiments were run over 5 independent trials. The plots give the averages and the standard deviations of the SNRs obtained at each iteration over the different trials. We observe that the proposed method is better than IPR, but the performance of INKSVD is superior to the other two methods.

However, we observe that each iteration of MOD-G is inexpensive compared to IPR and INKSVD. The 20 iterations of MOD-G algorithm took on an average 320 secs, averaged over the 10 (5+5) trials, to run on a 3.3 GHz Intel Core i5 machine executed using Matlab R2013a and using `cputime` function. The 20 iterations of IPR took a total of 1110 secs on average to run. IPR was over 3 times slower than MOD-G, this is because in each iteration of IPR, we need to compute an eigenvalue decomposition of a coherent constrained Gram matrix to threshold the eigenvalues, and also compute an SVD to rotate the dictionary. There are many additional matrix-matrix products in an IPR iteration compared to MOD-G. The 20 iterations of INKSVD took an average of 2067 secs. INK-SVD takes longer (almost 7 times longer than MOD-G) to compute less coherent dictionaries. This is because INK-SVD acts in a greedy fashion by decorrelating pair of atoms until the target mutual coherence is reached (or until a maximum number of iterations) and therefore the number of pairs of atoms to decorrelate increases for low values of the target coherence. For additional results and timing comparisons between IPR and INKSVD, see (Barchiesi and Plumbley, 2013). Hence, INKSVD gives a superior performance at the expense of higher per iteration runtime cost. The proposed DL algorithm produces comparable results, and is significantly faster (in terms of per iteration runtime) compared to the other two incoherent DL algorithms.

6 Conclusion

In this letter, we presented a new parallel (all atoms are updated simultaneously) dictionary learning algorithm. The proposed algorithm is based on adding a dictionary rank shrinkage step to improve the coherence of the

[3] SmallBox is an open source MatLab toolbox, containing codes and datasets for testing and benchmarking various dictionary learning algorithms `http://www.small-project.eu/software-data/smallbox/`.

dictionary at each iteration of the algorithm. This rank shrinkage step is applied on the least square estimate of the dictionary obtained from the dictionary update stage. In order to ensure that the decorrelated dictionary learned provides a good approximation the set of signals $\mathbf{Y}$, the rank shrinkage is achieved by transforming the problem of reducing the rank of the learned dictionary to a nonnegative garrotte estimation problem of the reshaped rank one decomposition of the dictionary. While in this paper we have focused on MOD to generate a more stable least square estimate, the proposed coherence reducing step can be accommodated with other dictionary learning methods as well, e.g., KSVD. Numerical experiments illustrated that the performance of the proposed method is comparable and many times better than other popular dictionary learning methods.

References

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54:4311–4322.

Bajwa, W. U., Calderbank, R., and Jafarpour, S. (2010). Why gabor frames? two fundamental measures of coherence and their role in model selection. *Communications and Networks, Journal of*, 12(4):289–307.

Barchiesi, D. and Plumbley, M. D. (2013). Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. *IEEE Transactions on Signal Processing*, 61:2055–2065.

Breiman, L. (1995). Best subset repression using the nonnegative garrotte. *Technometrics*, 37:373–384.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

Cleju, N. (2014). Optimized projections for compressed sensing via rank-constrained nearest correlation matrix. *Applied and Computational Harmonic Analysis*, 36(3):495–507.

Damnjanovic, I., Davies, M. E., and Plumbley, M. D. (2010). Smallbox-an evaluation framework for sparse representations and dictionary learning algorithms. In *Latent variable analysis and signal separation*, pages 418–425. Springer.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Elad, M. (2007). Optimized projections for compressed sensing. *Signal Processing, IEEE Transactions on*, 55(12):5695–5702.

Elad, M. (2010). *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer.

Engan, K., Aase, S. O., and Hakon-Husoy, J. (1999). Method of optimal directions for frame design. *IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, pages 2443–2446.

Friedman, J., Hastie, T., Hofling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1:302–332.

Hanif, M. and Seghouane, A. K. (2014). Maximum likelihood orthogonal dictionary learning. *IEEE Workshop on Statistical Signal Processing (SSP)*, pages 141–144.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, 5(2):248–264.

Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T., and Sejnowski, T. J. (2003). Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396.

Mailhe, B., Barchiesi, D., and Plumbley, M. D. (2012). INK-SVD: learning incoherent dictionaries for sparse representations. *In proceedings of IEEE International Conference on Acoustic Speech and signal Processing, ICASSP*, pages 3573–3576.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE.

Ramırez, I., Lecumberry, F., and Sapiro, G. (2009). Sparse modeling with universal priors and learned incoherent dictionaries. Technical report, Citeseer.

Ramirez, I., Lecumberry, F., and Sapiro, G. (2009). Universal priors for sparse modeling. *In proceedings of IEEE CAMSAP, 2009*, pages 197–200.

Razaviyayn, M., Tseng, H.-W., and Luo, Z.-Q. (2015). Computational intractability of dictionary learning for sparse representation. *arXiv preprint arXiv:1511.01776*.

Rubinstein, R., Bruckstein, A. M., and Elad, M. (2010). Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057.

Sahoo, S. K. and Makur, A. (2013). Dictionary training for sparse representation as generalization of K-means clustering. *IEEE Signal Processing Letters*, 20:587–590.

Schnass, K. and Vandergheynst, P. (2008). Dictionary preconditioning for greedy algorithms. *IEEE Transactions on Signal Processing*, 56(5):1994–2002.

Seghouane, A. K. (2010). Asymptotic bootstrap corrections of AIC for linear regression models. *Signal Processing*, 90:217–224.

Seghouane, A. K. and Hanif, M. (2015). A sequential dictionary learning algorithm with enforced sparsity. *In proceedings of IEEE International Conference on Acoustic Speech and signal Processing, ICASSP*, pages 3876–3880.

Tropp, J. A. (2004). Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242.

Tropp, J. A. (2008). On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25(1):1–24.

Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666.

Yuan, M. and Lin, Y. (2007). On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161.