COHERENCE REGULARIZED DICTIONARY LEARNING

Mansour Nejati[1], *Shadrokh Samavi*[1], *S. M. Reza Soroushmehr*[2,3], *Kayvan Najarian*[2,3,4]

[1]ECE Department, Isfahan University of Technology, Iran
[2]Department of Emergency Medicine, University of Michigan, Ann Arbor, USA
[3]University of Michigan Center for Integrative Research in Critical Care (MCIRCC), USA
[4]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, USA

ABSTRACT

Sparse representations over redundant learned dictionaries have shown to produce high quality results in various image processing tasks. An important characteristic of a learned dictionary is the mutual coherence of dictionary that affects its generalization performance and the optimality of sparse codes generated from it. In this paper, we present a dictionary learning model equipped with coherence regularization. For this model, two novel dictionary optimization algorithms based on group-wise minimization of inter- and intra-coherence penalties are proposed. Experimental results demonstrate that the proposed algorithms improve the generalization properties and sparse approximation performance of the trained dictionary compared to several incoherent dictionary learning methods.

Index Terms— Dictionary learning, mutual coherence, sparse representation, optimization, sparse coding.

1. INTRODUCTION

Sparse representation has proven to be an extremely powerful tool for analyzing a large class of signals. This is mainly due to the fact that important classes of signals have naturally sparse representations with respect to some dictionaries of basis elements referred to as dictionary atoms [1]. The dictionary plays a critical role in a successful sparse representation modeling and learned overcomplete dictionaries have become popular in recent years.

Dictionary learning for sparse representation has motivated a large amount of work over the past 20 years. This has led to state-of-the-art results in many signal and image processing tasks [2]. Learning methods mostly involve dictionary optimization in terms of the task to be performed. Data fitting and sparsity are two main objectives of optimization in many dictionary learning methods especially in reconstruction tasks. Moreover, imposing additional constraints or regularizations on certain intrinsic properties of dictionary can improve its performance [3].

One such important dictionary property is the mutual coherence, $\mu(\mathbf{D})$, of dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$ which measures the maximal correlation of any two distinct atoms in the dictionary and is defined as:

$$\mu(\mathbf{D}) \stackrel{\text{def}}{=} \max_{i \neq j} \left| \langle \mathbf{d}_i, \mathbf{d}_j \rangle \right| = \max_{i \neq j} \left| \mathbf{d}_i^T \mathbf{d}_j \right| \quad (1)$$

where $\mathbf{d}_i$ is the i-th atom (column) of $\mathbf{D}$, and $\langle .,. \rangle$ denotes the inner product operator. According to theoretical results on sparse coding, mutual coherence of dictionary has direct impact on stability and performance of coding algorithms [4], [5]. A lower coherence permits better sparse recovery. Furthermore, reducing the coherence of dictionary atoms can increase its generalization performance by reduction of over-fitting to the training data and avoiding atom degeneracy [6].

In this paper, we present a dictionary learning model equipped with a correlation penalty which induces low coherence dictionaries. To solve the dictionary optimization problem, we develop two new algorithms based on group-wise inter-group and intra-group coherence minimization. The experimental evaluations show the improved sparse reconstruction performance of trained dictionaries by our algorithms compared to other competing ones.

The rest of the paper is organized as follows. Section 2 briefly reviews related work on incoherent dictionary learning. In Section 3, we elaborate on the details of the proposed dictionary learning algorithms. Experimental results are presented in Section 4. Finally, we conclude the paper in Section 5.

2. RELATED WORK

There are some recent works on learning incoherent dictionaries for sparse representation [3], [6]-[9]. According to coherence reduction strategy, these approaches can be categorized into two groups. In the first group, incoherent dictionary learning is carried out by including a decorrelation step to an existing dictionary learning method. Methods presented in [7] and [8] are of this group both of which are based on K-SVD algorithm [10]. In [7], the coherence reduction is done via decorrelating pairs of atoms in a greedy way until the target coherence level is reached. However, data approximation error is not considered in the decorrelation step which can result in low coherence dictionaries with low sparse approximation performance. To cope with this problem, method of [8] applies decorrelation along with a rotation step to improve the approximation performance of the decorrelated dictionary. But, as shown in [8], this method does not work well for above-moderate coherence levels. In addition, the extra steps of decorrelation and rotation increase the total computation cost.

The second strategy to learn the incoherent dictionary is augmenting the dictionary learning objective with a coherence penalty. The methods presented in [3], [6], and [9] are based on this strategy. The coherence penalty term in all of these methods is the same but they use different approaches for minimization. In [3], an additional quadratic regularization term on atom's norm is added to learning model and the dictionary updates is performed by method of optimal coherence-constrained direction (MOCOD). In [6], several iterations of the limited-memory BFGS algorithm [11] are run for dictionary optimization. The same optimization problem is solved using gradient projection method in [9].

3. PROPOSED DICTIONARY LEARNING METHOD

In order to reduce the coherence of a trained dictionary, we present a **co**herence **re**gularized (CORE) dictionary learning model which explicitly imposes a regularizer on the coherence among dictionary atoms. Consider a training set of signals $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{n\times N}$ from which we want to learn a dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_K] \in \mathbb{R}^{n\times K}$. The proposed CORE learning model is given as follows:

$$\min_{\mathbf{D}\in\mathcal{D},\,\mathbf{X}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\sum_{i=1}^{N}\|\mathbf{x}_i\|_0 + \eta\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left(\mathbf{d}_i^T\mathbf{d}_j\right)^2 \quad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{K\times N}$ the sparse representation matrix and $\mathcal{D}$ is defined as the set of all dictionaries with unit ℓ_2-norm atoms. Also, $\|\cdot\|_0$ denotes ℓ_0-pseudo-norm that counts the number of non-zero entries. Here, the coherence penalty $\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left(\mathbf{d}_i^T\mathbf{d}_j\right)^2$ has been added to the reconstruction objective to promote the incoherence of dictionary atoms, where $\eta \geq 0$ is trade-off parameter. A classical alternate minimization scheme is used to deal with the problem in (2) by alternating between the sparse-coding and dictionary-update steps. In the sparse coding step, we adopt orthogonal matching pursuit (OMP) [12] because of its simplicity and efficiency. Our focus is on the dictionary update step in which we have to solve:

$$\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \eta\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\left(\mathbf{d}_i^T\mathbf{d}_j\right)^2 \quad (3)$$

We perform the dictionary update in a block coordinate fashion where one block of dictionary elements is optimized at a time while keeping the other ones fixed. Specifically, let $\mathbf{D}_\Omega = [\mathbf{d}_i]_{i\in\Omega}$ denote a subset of atoms indexed by $\Omega \subset \mathcal{I} = \{i\}_{i=1}^K$ that we want to update. Also, the rest of dictionary atoms are denoted by $\mathbf{D}_{\bar{\Omega}} = [\mathbf{d}_i]_{i\in\bar{\Omega}}$ with $\bar{\Omega} = \mathcal{I} \setminus \Omega$. Then, minimizing (3) with respect to $\mathbf{D}_\Omega$, when keeping $\mathbf{D}_{\bar{\Omega}}$ fixed, can be formulated as

$$\min_{\mathbf{D}_\Omega\in\mathcal{D}} \left\|\mathbf{Y}-\mathbf{D}_{\bar{\Omega}}\mathbf{X}_{[\bar{\Omega}]}-\mathbf{D}_\Omega\mathbf{X}_{[\Omega]}\right\|_F^2 + \eta\left\|\mathbf{D}_{\bar{\Omega}}^T\mathbf{D}_\Omega\right\|_F^2 + \frac{\eta}{2}\|\mathbf{D}_\Omega^T\mathbf{D}_\Omega-\mathbf{I}\|_F^2 \quad (4)$$

where $\mathbf{X}_{[\Omega]}$ indicates the sub-matrix of $\mathbf{X}$ consisting of its rows indexed by Ω, and $\mathbf{I}$ is the $|\Omega| \times |\Omega|$ identity matrix.

Let $\mathbf{E}_\Omega = \mathbf{Y}-\mathbf{D}_{\bar{\Omega}}\mathbf{X}_{[\bar{\Omega}]}$ be error matrix for all the N samples when the atoms indexed by Ω are removed from the dictionary. According to support of $\mathbf{x}_{[i]}$, the i-th row of $\mathbf{X}$, we define the set $\rho_i = \{j : 1 \leq j \leq N, \mathbf{x}_{[i]}(j) \neq 0\}$ containing the indices of samples $\{\mathbf{y}_j\}$ that use the atom $\mathbf{d}_i$. Accordingly, we define $\rho_\Omega = \bigcup_{i\in\Omega}\rho_i$ as the set of indices pointing to samples in $\mathbf{Y}$ that use at least one atoms of $\mathbf{D}_\Omega$ in their representation. Since $\mathbf{D}_\Omega$ has no impact on the representation error of samples $\{\mathbf{y}_j\}_{j\notin\rho_\Omega}$, we can replace $\mathbf{Y}$ and $\mathbf{X}$ in (4) by their column-reduced version $\acute{\mathbf{Y}} = [\mathbf{y}_j]_{j\in\rho_\Omega}$ and $\acute{\mathbf{X}} = [\mathbf{x}_j]_{j\in\rho_\Omega}$ respectively. Thus the objective function in (4) can be written as follows where the column-reduced error matrix is denoted by $\acute{\mathbf{E}}_\Omega$.

$$\min_{\mathbf{D}_\Omega} \left\|\acute{\mathbf{E}}_\Omega-\mathbf{D}_\Omega\acute{\mathbf{X}}_{[\Omega]}\right\|_F^2 + \eta\left\|\mathbf{D}_{\bar{\Omega}}^T\mathbf{D}_\Omega\right\|_F^2 + \frac{\eta}{2}\|\mathbf{D}_\Omega^T\mathbf{D}_\Omega-\mathbf{I}\|_F^2 \quad (5)$$

This dictionary update model has two coherence penalization terms. The term $\left\|\mathbf{D}_{\bar{\Omega}}^T\mathbf{D}_\Omega\right\|_F^2$ is inter-coherence penalty that promotes incoherence between atoms of $\mathbf{D}_\Omega$ and the rest of the dictionary atoms. The other coherence penalty term suppresses the intra-coherence of atoms in $\mathbf{D}_\Omega$.

3.1. CORE-I learning model

We first consider the dictionary update model in (5) regardless of intra-coherence penalty and instead encourage low intra-coherence by random selection of atom subsets at each dictionary update step. We refer to the resulting update model with inter-coherence regularization as CORE-I. Taking the derivative of the related objective function with respect to $\mathbf{D}_\Omega$ and setting it equal to zero results in:

$$\eta\mathbf{D}_{\bar{\Omega}}\mathbf{D}_{\bar{\Omega}}^T\mathbf{D}_\Omega + \mathbf{D}_\Omega\acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T \quad (6)$$

Let us define matrices $\mathbf{A} = \eta\mathbf{D}_{\bar{\Omega}}\mathbf{D}_{\bar{\Omega}}^T$, $\mathbf{B} = \acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T$, and $\mathbf{C} = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T$. Using these definitions, (6) is reduced to the following matrix equation.

$$\mathbf{A}\mathbf{D}_\Omega + \mathbf{D}_\Omega\mathbf{B} = \mathbf{C} \quad (7)$$

The above matrix equation is known as the Sylvester equation. This matrix equation and Lyapunov matrix equation as a special case ($\mathbf{B} = \mathbf{A}^T$) are very important equations in theory and applications and arise in many diverse engineering and mathematics problems [13], [14].

The most widely used standard method for numerical solution of the Sylvester equation is the well-known Bartels-Stewart method [15]. The main idea of the Bartels-Stewart method is to transform matrices $\mathbf{A}$ and $\mathbf{B}$ into triangular form by orthogonal similarity transformation and then solving the resulting transformed system by back-substitution. We make use of this method in our dictionary update problem to calculate the solution of (7) and thus to update $\mathbf{D}_\Omega$. This update procedure is followed by normalizing the columns of $\mathbf{D}_\Omega$ to have unit ℓ_2-norm.

Moreover, along with updating the atoms of $\mathbf{D}_\Omega$, we also update the nonzero coefficients in their associated row vectors in $\mathbf{X}$. This can accelerate convergence, since more relevant coefficients will be used in the subsequent atom updates. Let $\omega_i = \{j : 1 \le j \le K, \mathbf{x}_i(j) \neq 0\}$ be the index set of nonzero entries in $\mathbf{x}_i$, the *i*-th column of $\mathbf{X}$. Updating the nonzero sparse coefficients associated to $\mathbf{D}_\Omega$ involves solving a least squares problem for each column of $\acute{\mathbf{X}}_{[\Omega]}$ as:

$$\min_{\mathbf{x}_i(\Omega\cap\omega_i)} \left\|\mathbf{e}_{\Omega,i} - \mathbf{D}_{\Omega\cap\omega_i}\mathbf{x}_i(\Omega\cap\omega_i)\right\|_2^2 \quad ,\forall i \in \rho_\Omega \qquad (8)$$

where $\mathbf{e}_{\Omega,i}$ is the *i*-th column of the error matrix $\mathbf{E}_\Omega$. This problem admits a closed-form solution thereby the coefficients update rule is given by

$$\mathbf{x}_i(\Omega\cap\omega_i) \leftarrow \left(\mathbf{D}_{\Omega\cap\omega_i}^T\mathbf{D}_{\Omega\cap\omega_i}\right)^{-1}\mathbf{D}_{\Omega\cap\omega_i}^T\mathbf{e}_{\Omega,i} \quad ,\forall i \in \rho_\Omega \qquad (9)$$

Once the updating of all subsets of dictionary atoms and their corresponding nonzero coefficients is done, next iteration of dictionary learning algorithm is started by sparse coding step. Algorithm 1 presents the overall procedure of the proposed CORE-I dictionary update. By $\mathrm{Sylv}(\mathbf{D}_\Omega;\mathbf{A},\mathbf{B},\mathbf{C})$ we mean the solving of matrix Sylvester equation in (7) using the Bartels-Stewart algorithm.

3.2. CORE-II learning model

We return to our dictionary optimization problem defined in (5) again but this time we consider both inter- and intra-incoherence inducing terms. To obtain the updated dictionary, the derivative of optimization function with respect to $\mathbf{D}_\Omega$ is set to zero and then is rearranged as:

$$\eta\mathbf{D}_{\bar\Omega}\mathbf{D}_{\bar\Omega}^T\mathbf{D}_\Omega + \mathbf{D}_\Omega\left(\acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T + \eta\mathbf{D}_\Omega^T\mathbf{D}_\Omega - \eta\mathbf{I}_m\right) = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T \qquad (10)$$

Defining the matrices $\mathbf{A} = \eta\mathbf{D}_{\bar\Omega}\mathbf{D}_{\bar\Omega}^T$, $\mathbf{B} = \acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T + \eta\mathbf{D}_\Omega^T\mathbf{D}_\Omega - \eta\mathbf{I}_m$, and $\mathbf{C} = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T$, we again achieve a matrix equation of the form (7). In this case, however, the coefficient matrix $\mathbf{B}$ depends on $\mathbf{D}_\Omega$. Therefore, we apply an iterative approach for updating $\mathbf{D}_\Omega$ as

$$\mathbf{D}_\Omega^{(t+1)} \leftarrow \mathrm{Sylv}\left(\mathbf{D}_\Omega^{(t)};\mathbf{A},\mathbf{B}^{(t)},\mathbf{C}^{(t)}\right) \qquad (11)$$

which is followed by updating the corresponding nonzero coefficients using (9). The updated $\mathbf{D}_\Omega$ and nonzero coefficients at each iteration are used to recalculate the matrices $\mathbf{B}$ and $\mathbf{C}$. In practice, we found that a few (e.g. 4) iteration is sufficient to give the appropriate updates. This dictionary update method referred to as CORE-II is summarized in Algorithm 2.

Algorithm 1 CORE-I Dictionary Update

Require: Training set $\mathbf{Y} = [\mathbf{y}_j]_{j=1}^N$ in $\mathbb{R}^{n\times N}$, dictionary $\mathbf{D} \in \mathcal{D}$, sparse coefficients $\mathbf{X} \in \mathbb{R}^{K\times N}$, regularization parameter η, number of atoms in each subset m

Initialization: $\Omega_0 = \{1,2,\dots,K\}$.
while $\Omega_0 \neq \emptyset$ **do**
  $\Omega \leftarrow$ Pick m index from Ω_0 randomly.
  $\bar\Omega \leftarrow \{1 \le j \le K : j \notin \Omega\}$.
  $\Omega_0 \leftarrow \Omega_0 \setminus \Omega$.
  $\rho_\Omega = \cup_{i\in\Omega}\{j : 1 \le j \le N, \mathbf{x}_{[i]}(j) \neq 0\}$.
  $\acute{\mathbf{Y}} = [\mathbf{y}_j]_{j\in\rho_\Omega}$, $\acute{\mathbf{X}} = [\mathbf{x}_j]_{j\in\rho_\Omega}$.
  $\acute{\mathbf{E}}_\Omega = \acute{\mathbf{Y}} - \mathbf{D}_{\bar\Omega}\acute{\mathbf{X}}_{[\bar\Omega]}$.
  $\mathbf{A} = \eta\mathbf{D}_{\bar\Omega}\mathbf{D}_{\bar\Omega}^T$, $\mathbf{B} = \acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T$, $\mathbf{C} = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T$.
  $\mathbf{D}_\Omega \leftarrow \Pi_{\mathcal{D}}(\mathrm{Sylv}(\mathbf{D}_\Omega;\mathbf{A},\mathbf{B},\mathbf{C}))$.
  Update nonzero coefficients in $\acute{\mathbf{X}}_{[\Omega]}$ via (9).
end while
return D (updated dictionary)

Algorithm 2 CORE-II Dictionary Update

Require: Training set $\mathbf{Y} = [\mathbf{y}_j]_{j=1}^N$ in $\mathbb{R}^{n\times N}$, dictionary $\mathbf{D} \in \mathcal{D}$, sparse coefficients $\mathbf{X} \in \mathbb{R}^{K\times N}$, regularization parameter η, number of atoms in each subset m

Initialization: $\Omega_0 = \{1,2,\dots,K\}$.
while $\Omega_0 \neq \emptyset$ **do**
  $\Omega \leftarrow$ Pick m index from Ω_0 randomly.
  $\bar\Omega \leftarrow \{1 \le j \le K : j \notin \Omega\}$.
  $\Omega_0 \leftarrow \Omega_0 \setminus \Omega$.
  $\rho_\Omega = \cup_{i\in\Omega}\{j : 1 \le j \le N, \mathbf{x}_{[i]}(j) \neq 0\}$.
  $\acute{\mathbf{Y}} = [\mathbf{y}_j]_{j\in\rho_\Omega}$, $\acute{\mathbf{X}} = [\mathbf{x}_j]_{j\in\rho_\Omega}$.
  $\acute{\mathbf{E}}_\Omega = \acute{\mathbf{Y}} - \mathbf{D}_{\bar\Omega}\acute{\mathbf{X}}_{[\bar\Omega]}$.
  $\mathbf{A} = \eta\mathbf{D}_{\bar\Omega}\mathbf{D}_{\bar\Omega}^T$.
  for $t = 1,\dots,T$ **do**
    $\mathbf{B} = \acute{\mathbf{X}}_{[\Omega]}\acute{\mathbf{X}}_{[\Omega]}^T + \eta\mathbf{D}_\Omega^T\mathbf{D}_\Omega - \eta\mathbf{I}_m$, $\mathbf{C} = \acute{\mathbf{E}}_\Omega\acute{\mathbf{X}}_{[\Omega]}^T$.
    $\mathbf{D}_\Omega \leftarrow \Pi_{\mathcal{D}}(\mathrm{Sylv}(\mathbf{D}_\Omega;\mathbf{A},\mathbf{B},\mathbf{C}))$.
    Update nonzero coefficients in $\acute{\mathbf{X}}_{[\Omega]}$ via (9).
  end for
end while
return D (updated dictionary)

4. EXPERIMENTAL RESULTS

In this section, the performance of the proposed coherence regularized dictionary learning algorithms is evaluated[1]. We compare CORE methods (CORE-I and CORE-II) with several recently proposed incoherent dictionary learning algorithms. These algorithms are incoherent K-SVD (INK-SVD) [7], K-SVD decorrelation with iterative projections and rotations (IPR) [8], incoherent dictionary learning based on BFGS algorithm (IDL-BFGS) [6], and method of optimal coherence-constrained directions (MOCOD) [3]. Results of

[1] The code is available at: http://mansournejati.ece.iut.ac.ir/content/core-dl

these methods are obtained using the source codes provided by their authors and all experiments are performed using MATLAB R2012b on a PC with 3.4 GHz Intel Core-i7 CPU and 8 GB RAM.

We take a collection of 100 images from the Berkeley segmentation dataset[2] to form the training set consisting of 80000 randomly chosen 8×8 patches. A set of 80000 test image patches are also extracted from the remaining images of this dataset. The dictionary size is set to be $K = 512$ atoms and we run all dictionary learning algorithms for 50 iterations. Also, in order to have a fair comparison among different incoherent dictionary updates, OMP [12] is used in the sparse coding step of all competing algorithms.

Regularization parameter η and cardinality of subsets of dictionary atoms m, are two important parameters of CORE methods needed to be tuned for optimality. Based on our experiments, we found that increasing m to about half of the patch size improves the reconstruction performance. However, setting m to the larger values either does not improve the performance or even degrades it. Therefore, in our experiments, we set m to $\frac{n}{2}$ where n denotes the patch size. The regularization parameter η is tuned heuristically to produce the best results in terms of sparse reconstruction performance. The coherence related parameters of the other competing methods are also chosen such that the trained dictionaries have similar mutual coherence.

We apply the trained dictionaries to sparse approximation of test image patches using OMP. The reconstruction performance of dictionary is evaluated using signal to noise ratio (SNR) of these approximations as:

$$\mathrm{SNR} = 10 \log_{10} \frac{\|\mathbf{Y}\|_F^2}{\|\mathbf{Y} - \mathbf{DX}\|_F^2} \qquad (12)$$

where $\mathbf{Y}$ and $\mathbf{X}$ denote the test set and their sparse representations respectively. Figure 1 shows the SNR of the reconstructions associated with various dictionary learning algorithms as a function of average number of atoms required. It can be seen that the dictionaries trained by our algorithms have better reconstruction SNR which implies the improved generalization performance of the dictionaries trained by our algorithms. Another observation is that performances of both CORE methods are similar. In fact, CORE-II leads to slightly more reconstruction SNR for sparsity levels beyond 5. This can be due to the iterative application of (11) for each sub-dictionary which can result in more adapted dictionary atoms.

Table 1 presents a detailed comparison where $\mu_{\text{avg}} = \frac{1}{K(K-1)} \sum_i \sum_{j \neq i} |\mathbf{d}_i^T \mathbf{d}_j|$ denotes the average mutual coherence and the reconstruction SNR is reported for average sparsity level of 8. It can be seen that for the same value of μ (all trained dictionary have $\mu = 0.94$), CORE algorithms lead to the lowest μ_{avg} and the highest SNR as compared to other incoherent dictionary learning algorithms. The closest competitor to CORE methods in SNR performance is IDL-BFGS [6] which is more than 3 times slower. In terms of execution time, the CORE algorithms are much faster than IPR [8], IDL-BFGS [6], and INK-SVD [7]. Only MOCOD [3] has smaller execution time than CORE algorithms but this method yields the worst SNR performance among the others. It is also observed that CORE-II leads to lower μ_{avg} than CORE-I. This is because CORE-II optimizes the dictionary with both inter and intra coherence suppressing terms. However, this lower coherence is obtained at the cost of an increased run time.

5. CONCLUSION

In this paper, we presented a coherence regularized dictionary learning model to jointly minimize the data approximation error and the coherence of dictionary atoms. We considered the group-wise simultaneous updating of dictionary atoms and propose two novel algorithms for solving the corresponding optimization problem based inter- and intra-coherence minimization. The evaluations we carried out show that our algorithms improve the sparse approximation performance of trained dictionaries compared to previous approaches to incoherent dictionary learning, while requiring a lower computational time.

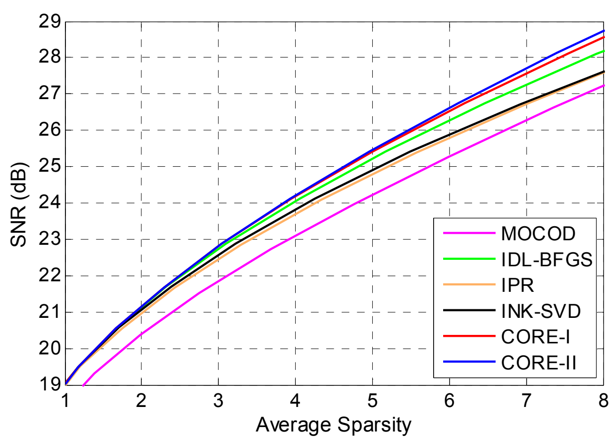

Fig. 1. Generalization performance of dictionaries trained by CORE learning methods compared with competing algorithms.

Table 1. Comparison of different dictionary learning algorithms in terms of average mutual coherence of trained dictionary, sparse reconstruction performance on test set, and learning run time.

Algorithm	μ_{avg}	SNR (dB)	Run Time (s)
CORE-I	0.1080	28.56	149
CORE-II	0.0919	28.73	201
INK-SVD [7]	0.1915	27.62	402
IPR [8]	0.2169	27.59	731
MOCOD [3]	0.1388	27.23	120
IDL-BFGS [6]	0.1258	28.18	608

[2] www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

6. REFERENCES

[1] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, June 2010.

[2] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing", *Foundations and Trends® in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2012.

[3] I. Ramirez, F. Lecumberry, and G. Sapiro, Sparse modeling with universal priors and learned incoherent dictionaries, Tech. Rep., IMA, University of Minnesota, 2009.

[4] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[5] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[6] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Learning dictionaries with bounded self-coherence," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 861–864, Dec. 2012.

[7] B. Mailhé, D. Barchiesi, and M. D. Plumbley, "INK-SVD: Learning incoherent dictionaries for sparse representations," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3573–3576.

[8] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2055–2065, 2013.

[9] K. Balasubramanian, K. Yu, and G. Lebanon, "Smooth sparse coding via marginal regression for learning sparse representations," *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013.

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

[11] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, 1989.

[12] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, pp. 40–44.

[13] Z. Gajic and M. T. J. Qureshi, *Lyapunov matrix equation in system stability and control*. Courier Corporation, 2008.

[14] G. W. Stewart, *Matrix Algorithms Volume 2: Eigensystems*. SIAM, 2001.

[15] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation AX+ XB= C," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, Sept. 1972.