Dictionary Learning with Low Mutual Coherence Constraint

Mostafa Sadeghi, Massoud Babaie-Zadeh*

Electrical Engineering Department, Sharif University of Technology, Tehran, Iran

Abstract

This paper presents efficient algorithms for learning low-coherence dictionaries. First, a new algorithm based on proximal methods is proposed to solve the dictionary learning (DL) problem regularized with the mutual coherence of dictionary. This is unlike the previous approaches that solve a regularized problem where an approximate incoherence promoting term, instead of the mutual coherence, is used to encourage low-coherency. Then, a new solver is proposed for constrained low-coherence DL problem, i.e., a DL problem with an explicit constraint on the mutual coherence of the dictionary. As opposed to current methods, which follow a suboptimal two-step approach, the new algorithm directly solves the associated DL problem. Using previous studies, convergence of the new schemes to critical points of the associated cost functions is also provided. Furthermore, it is shown that the proposed algorithms have lower iteration complexity than existing algorithms. Our simulation results on learning low-coherence dictionaries for natural image patches as well as image classification based on discriminative over-complete dictionary learning demonstrate the superiority of the proposed algorithms compared with the state-of-the-art method.

Keywords: Sparse approximation, dictionary learning, mutual coherence, proximal mapping, penalty method

*Corresponding author

1. Introduction

1.1. Sparse signal approximation

Sparsity has been a key concept in a wide range of signal processing and machine learning problems over the last decade [1]. In particular, sparse signal approximation has been extensively utilized in a variety of applications, including image enhancement [2, 1]. To be more precise, let $\mathbf{y} \in \mathbb{R}^n$ denote the target signal, and $\{\mathbf{d}_i\}_{i=1}^N$ be a number of N atoms collected as the columns of a so-called *dictionary* $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_N]$. For more flexibility, the dictionary is usually chosen to be overcomplete, *i.e.*, $N > n$. Then, the approximation of $\mathbf{y}$ over $\mathbf{D}$ is written as $\mathbf{y} \approx \sum_{i=1}^N x_i \mathbf{d}_i = \mathbf{D}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^N$ is called the sparse representation vector, with most of its entries being zero. The sparse approximation problem is to find the sparsest $\mathbf{x}$. To this end, the following problem has to be solved:

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon, \tag{1}$$

where $\|.\|_0$, the so-called ℓ_0 (pseudo) norm, returns the number of non-zero entries, and $\epsilon \geq 0$ is an error tolerance. Various sparse recovery algorithms have been proposed, which are summarized in [3].

1.2. Traditional dictionary learning

Dictionary has an important role in sparse approximation problems. There exist some predefined choices for the dictionary, including discrete cosine transform (DCT) for natural images and Gabor dictionaries for speech signals [4]. Nevertheless, it has been shown that *learned dictionaries* optimized over a set of training signals outperform predefined ones in many applications [1, 4, 5, 6, 7]. This process is known as *dictionary learning* (DL) [4]. In DL, given a training data matrix $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_M]$, a dictionary $\mathbf{D}$ is learned from $\mathbf{Y}$, in such a way that it provides sparse enough representations for $\mathbf{y}_i$'s. This task can be formulated as the following problem

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2, \tag{2}$$

where $\|.\|_F$ is the Frobenius norm, and $\mathcal{D}$ and $\mathcal{X}$ are admissible sets of $\mathbf{D}$ and $\mathbf{X}$, respectively. $\mathcal{D}$ is usually defined as $\mathcal{D} \triangleq \left\{\mathbf{D} \in \mathbb{R}^{n\times N} \mid \ \forall i, \|\mathbf{d}_i\|_2 = 1\right\}$, which will be also considered in this paper. Furthermore, the set $\mathcal{X}$ constrains $\mathbf{X}$ to have sparse columns.

Numerous DL algorithms have been proposed [8], which follow an alternating minimization approach to solve the DL problem in (2). That is, starting with an initial estimate for the dictionary, most DL solvers alternate between the following two stages:

1. **Sparse approximation (SA)**: The training signals are sparsely approximated over the current estimate of $\mathbf{D}$.
2. **Dictionary update (DU)**: The dictionary is updated by minimizing the approximation error of the SA stage.

1.3. Low-coherence dictionary learning

To ensure computational tractability and successful performance of sparse approximation algorithms, the dictionary must satisfy certain properties. In fact, substantial efforts in investigating the theoretical aspect of the sparse approximation problem have revealed that the uniqueness and stability of sparse approximation are directly related to the dictionary [9, 10]. Mutual coherence (MC) [11] and restricted isometry property (RIP) [9] are two fundamental tools for evaluating the goodness of a dictionary. The MC for a dictionary $\mathbf{D}$ with normalized columns is defined as

$$\mu(\mathbf{D}) \triangleq \max_{i\neq j} \ |\langle \mathbf{d}_i, \mathbf{d}_j\rangle|\cdot \tag{3}$$

MC simply measures the similarity between distinct atoms of a dictionary. For $n \times N$ dictionaries, the MC is lower bounded via

$$\mu \geq \sqrt{\frac{N-n}{n(N-1)}}, \tag{4}$$

which is known as the Welch bound [12]. In addition, $\mathbf{D}$ is said to satisfy RIP of order s with constant δ_s if for any s-sparse signal $\mathbf{x}$ we have [13]

$$(1-\delta_s)\|\mathbf{x}\|_2^2 \leq \|\mathbf{D}\mathbf{x}\|_2^2 \leq (1+\delta_s)\|\mathbf{x}\|_2^2\cdot \tag{5}$$

The existing RIP-based performance guarantees for sparse approximation algorithms show that smaller values for δ_s are favorable. Intuitively, according to (5), when $\delta_s \to 0$ the dictionary behaves as an orthonormal basis and, so, the stability and uniqueness of the approximation are better satisfied. Evaluating the RIP for a dictionary, however, is an NP-hard problem [14]. Nevertheless, it can be roughly verified using MC. In fact, it has been shown that RIP and MC for a dictionary are linked via $\delta_s \leq \mu(s-1)$ [13]. Furthermore, it can be shown that $\delta_2 = \mu$ [1]. Consequently, MC as a practical measure of performance has received much attention [15, 16, 17].

Besides the sole sparse approximation problem, recent work indicates that MC and RIP play effective roles in theoretical investigations of the DL problem as a whole. For instance, it is argued in [18, 19] that a ground truth dictionary would be a local minimum to the DL cost function with an ℓ_1 norm as the sparsity measure, provided that it is sufficiently incoherent along with some other assumptions on the problem parameters. In this case, the dictionary is said to be *locally identifiable* [18]. Moreover, Wu and Yu [19] showed recently that, under some conditions, the local identifiability is possible with sparsity level s to the order $\mathcal{O}(\mu^{-2})$ for a complete dictionary ($n = N$) with the MC value of μ. Also, in [20], RIP has been used as a determining assumption in providing local linear convergence for the alternating minimization approach used to solve the DL problem.

The importance of MC, explained by the above discussions, has inspired some work to propose algorithms for learning low-MC dictionaries. These work can be classified into two main groups: regularized and constrained approaches. The first group [21, 22, 23] targets the following problem to update the dictionary

$$\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\mathcal{R}(\mathbf{D}), \tag{6}$$

where $\lambda \geq 0$ is a regularization parameter, and

$$\mathcal{R}(\mathbf{D}) \triangleq \|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_F^2. \tag{7}$$

The use of $\mathcal{R}$ as an incoherence promoting term is motivated by the following

relation

$$\mathcal{R}(\mathbf{D}) = \sum_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|^2 + \sum_i (\langle \mathbf{d}_i, \mathbf{d}_i \rangle - 1)^2, \tag{8}$$

where the first term is responsible for minimizing the average squared inner-products between distinct atoms, and the last term encourages the atoms to have unit norms.

The second group of methods aims at solving the following constrained problem

$$\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0, \tag{9}$$

in which, $\mu_0 > 0$ is a fixed target MC level. Incoherent K-SVD (INK-SVD) [24] and iterative projection-rotation DL (IPR-DL) [25] are sample algorithms of this group. INK-SVD solves (9) by first finding the minimizer (denoted by $\bar{\mathbf{D}}$) ignoring the MC constraint and, then, solving the following matrix nearness problem to reach the desired mutual coherence:

$$\min_{\mathbf{D}} \frac{1}{2} \|\mathbf{D} - \bar{\mathbf{D}}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0. \tag{10}$$

To solve (10), the authors of [24] proposed a decorrelation step in which sub-dictionaries of highly correlated atoms are iteratively identified and pairs of atoms are decorrelated until the desired MC, namely μ_0, is reached. A disadvantage of this approach, as pointed out in [25], is that, the approximation error, *i.e.*, $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F$, is not explicitly taken into account during the decorrelation process. To overcome this problem, Barchiesi *et al.* [25] proposed a novel decorrelation scheme, consisting of two steps. In the first step, called *atoms decorrelation*, an iterative projection algorithm is performed that ensures that the mutual coherence constraint is satisfied. This step is then followed by a *dictionary rotation* in which the dictionary is rotated to minimize the approximation error, without affecting its mutual coherence. This algorithm shows state-of-the-art performance, as confirmed by the simulations of [25].

1.4. Our contributions

In this paper, we propose new algorithms for learning low-coherence dictionaries. Our motivation is that the previous work cannot efficiently compromise

between the representation ability of the learned dictionary and its MC. More precisely, as will be seen later in Section 2, the term $\mathcal{R}$ used in (6) is a very rough approximation to MC. In fact, as its definition in (8) implies, using this term only the *average* squared inner-products between distinct atoms are penalized. Therefore, it is not particularly effective to minimize the maximum absolute inner-products between any two atoms, i.e., the MC. Furthermore, when $\lambda \to \infty$, problem (6) reduces to

$$\min_{\mathbf{D}\in\mathcal{D}} \ \mathcal{R}(\mathbf{D}), \tag{11}$$

whose solutions are unit-norm tight-frames (UNTFs) [26, 27] which are not guaranteed to have small enough MCs compared to what one would expect in this extreme scenario. In other words, a UNTF may have a large MC. So, one would need to further optimize it to achieve a small enough MC [28].

To address this issue, we introduce a new algorithm based on proximal methods [29] to solve the regularized (unconstrained) low-coherence DL problem that directly uses the MC definition in (3) instead of the approximate term $\mathcal{R}$. Our simulations reveal that the new algorithm is able to learn dictionaries with MCs far smaller than what the algorithms targeting problem (6) achieve. Additionally, it makes much better compromise between adapting the dictionary to the training set and minimizing the MC.

On the other hand, IPR-DL, as the state-of-the-art algorithm, does not solve (9) directly. Moreover, singular value decomposition (SVD) and eigenvalue decomposition (EVD) are used in its structure, which are expensive operations from computational load and memory usage points of view, especially in high-dimensional data settings. Our next algorithm solves the constrained low-coherence DL problem (9) directly, without resorting to a suboptimal approach as in INK-SVD and IPR-DL. In addition, the proposed algorithm does not use SVD or EVD. As will be seen in Section 5, compared with IPR-DL, our proposed unconstrained algorithm is able to efficiently minimize the approximation error while satisfying the target level of MC.

Our proposed regularized low-coherence DL algorithm has already been in-

troduced in a conference paper [30]. However, the algorithm proposed in the current work is different from the one presented in [30]. More precisely, in [30] an atom-by-atom approach is taken that updates the atoms sequentially. However, in this work, we consider updating all the atoms at once. Moreover, here, we consider the constrained low-coherence DL problem, too, and in addition to presenting a new solver for the constrained case, we provide convergence guarantees for the dictionary update steps of both regularized and constrained problems based on previous studies.

1.5. Organization of the paper

The rest of the paper is presented in the following order. In Section 2, our new regularized low-coherence DL algorithm is introduced. The proposed constrained low-coherence DL algorithm is introduced in Section 3. Some discussions concerning the implementations of our new algorithms and a computational complexity analysis are given in Section 4. Simulation results will be reported in Section 5.

1.6. Notations and preliminaries

Throughout the paper, small and capital bold face characters are used for vector- and matrix-valued quantities, respectively. The (i, j)-th entry of a matrix $\mathbf{X}$ is denoted by x_{ij}, while $\mathbf{x}_i$ designates its i-th column. The superscript T stands for matrix transposition. The identity matrix is denoted by $\mathbf{I}$. For a vector $\mathbf{x}$, its ℓ_p norm ($p \geq 1$) is defined via $\|\mathbf{x}\|_p \triangleq (\sum_i |x_i|^p)^{1/p}$. For a matrix $\mathbf{X}$, we denote its ℓ_∞ norm[1] by $\|\mathbf{X}\|_\infty$ and define it as $\|\mathbf{X}\|_\infty \triangleq \max_{i,j} |x_{ij}|$. An ℓ_p norm-ball of radius r in $\mathbb{R}^n$ and centered around the origin is defined as $\mathcal{B}_p^r \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_p \leq r\}$. The same notation is used for ℓ_p matrix norm-ball. For two vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^n$, their inner-product is represented as $\langle \mathbf{u}, \mathbf{v} \rangle \triangleq \sum_i u_i v_i$. The indicator function of a set $\mathcal{S}$ is denoted by $\mathcal{I}_{\mathcal{S}}(\mathbf{x})$ which

[1] Note that the definition given here is different from the usual ℓ_∞ (induced) matrix norm, defined as the maximum absolute row sum of a matrix.

takes a value of 0 when $\mathbf{x} \in \mathcal{S}$, and ∞ otherwise. The vectorization operator is denoted by vec(.), which converts its matrix argument to an equivalent column vector by stacking its columns on top of each other. The inverse vectorization operator is also denoted by $\text{vec}^{-1}(.)$. Moreover,

Definition 1 ([29]). *The Euclidean projection of a point* $\mathbf{x} \in \mathbb{R}^n$ *to a non-empty set* $\mathcal{S} \subseteq \mathbb{R}^n$ *is defined as*

$$P_{\mathcal{S}}(\mathbf{x}) \triangleq \underset{\mathbf{u}\in\mathcal{S}}{\text{argmin}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2.$$

Definition 2 ([29]). *The proximal mapping of a convex function* $g: \ \text{dom}g \longrightarrow \mathbb{R}$ *is defined as*

$$prox_g(\mathbf{x}) \triangleq \underset{\mathbf{u}\in\text{dom}g}{\text{argmin}} \ \left\{\frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 + g(\mathbf{u})\right\}.$$

For $g(\mathbf{x}) = \mathcal{I}_{\mathcal{S}}(\mathbf{x})$, the proximal mapping is simply the projection onto $\mathcal{S}$ [29]. That is,

$$\text{prox}_{\mathcal{I}_{\mathcal{S}}}(\mathbf{x}) = P_{\mathcal{S}}(\mathbf{x}). \tag{12}$$

2. Proposed regularized algorithm

In this section, a new regularized low-coherence DL algorithm, called RINC-DL, for regularized incoherent DL, is introduced which tries to minimize the general DL cost function augmented with the MC term. To start, notice the following equivalent definition of MC for unit-norm dictionaries

$$\mu(\mathbf{D}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty. \tag{13}$$

Using this formula, the regularized problem that we target is expressed as

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_\infty. \tag{14}$$

Note that the regularization term does not depend on the coefficient matrix $\mathbf{X}$. Consequently, the SA stage in the general DL problem remains unchanged and any sparse approximation algorithm can be employed. So, let us focus on the DU stage

$$\min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_\infty. \tag{15}$$

Notice the difference between (13) and (7). It is observed that the regularization term $\mathcal{R}$ used in previous work is an approximation to MC in which the ℓ_∞ norm has been replaced with the Frobenius norm.

Similarly to (6), problem (15) is non-convex. However, solving (15) is more challenging due to the non-smoothness of the ℓ_∞ norm, which makes it difficult to directly apply optimization algorithms such as steepest descent. To circumvent this difficulty, we utilize proximal mappings. To reach this goal, an auxiliary variable is first defined: $\mathbf{G} \triangleq \mathbf{D}^T\mathbf{D}$. Then, problem (15) is reformulated as

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{G}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{G}-\mathbf{I}\|_\infty \quad \text{s.t.} \quad \mathbf{G}=\mathbf{D}^T\mathbf{D}. \tag{16}$$

By this trick, the quadratic term $\mathbf{D}^T\mathbf{D}$ is taken out of the non-smooth part of the objective function, which facilitates application of proximal algorithms [29]. We solve the above equality-constrained problem using penalty methods [31]. To this end, the following problem has to be solved

$$\min_{\mathbf{D},\mathbf{G}} \ \left\{ H(\mathbf{D},\mathbf{G}) \triangleq F(\mathbf{D},\mathbf{G}) + r_d(\mathbf{D}) + r_g(\mathbf{G}) \right\}, \tag{17}$$

in which,

$$F(\mathbf{D},\mathbf{G}) \triangleq \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \frac{1}{2\alpha}\|\mathbf{G}-\mathbf{D}^T\mathbf{D}\|_F^2, \tag{18}$$

$\alpha > 0$ is a penalty parameter, $r_d \triangleq \mathcal{I}_\mathcal{D}$ is the indicator function of $\mathcal{D}$, and $r_g(\mathbf{G}) \triangleq \lambda\|\mathbf{G}-\mathbf{I}\|_\infty$. When $\alpha \to 0$, the constraint violations are penalized with increasing severity and the solution of (17) approaches that of (16). Similar to a general penalty method, we consider a sequence $\{\alpha_i\}$ with $\alpha_i \to 0$, and find approximate minimizers of (17) for each i. The obtained minimizers, corresponding to α_i, are then used as initial points to perform the minimization corresponding to α_{i+1}. It should be mentioned that if the exact global minimizers are computed for each i, then every limit point of the sequence of global minimizers is a solution of (16), and thus, the original problem (15) (see Theorem 17.1 of [31]).

In order to solve (17) for each α_i, an alternating minimization approach is employed. In this way, the cost function is iteratively minimized, each time over

one variable while the other is fixed. The update problems for $\mathbf{D}$ and $\mathbf{G}$ are discussed in the following subsections.

2.1. Updating $\mathbf{G}$

After simple rearrangements, the update problem for $\mathbf{G}$ is converted to[2]

$$\mathbf{G}_{k+1} = \mathbf{I} + \underset{\mathbf{G}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{G} - \mathbf{M}_k\|_F^2 + \eta\|\mathbf{G}\|_\infty \right\}, \tag{19}$$

where $\mathbf{M}_k \triangleq \mathbf{D}_k^T\mathbf{D}_k - \mathbf{I}$, and $\eta \triangleq \lambda \cdot \alpha$. The second term in the right-hand side of (19) is, by definition, the proximal mapping of the ℓ_∞ matrix norm, which has a closed-form solution characterized by the following lemma.

Lemma 1. *Let g denote the function $\eta\|.\|_\infty : \mathbb{R}^{N\times N} \to \mathbb{R}$. The proximal mapping of g is given by*

$$prox_g(\mathbf{U}) = \mathbf{U} - vec^{-1}(P_{\mathcal{B}_1^\eta}(vec(\mathbf{U}))), \tag{20}$$

where, $P_{\mathcal{B}_1^\eta}(.) : \mathbb{R}^{N^2\times 1} \to \mathbb{R}^{N^2\times 1}$ is the projection onto the ℓ_1 norm-ball of radius η.

Proof: See Appendix A.

So, the update formula for $\mathbf{G}$ is

$$\mathbf{G}_{k+1} = \mathbf{I} + \mathbf{M}_k - \mathrm{vec}^{-1}(P_{\mathcal{B}_1^\eta}(\mathrm{vec}(\mathbf{M}_k))), \tag{21}$$

or equivalently,

$$\mathbf{G}_{k+1} = \mathbf{D}_k^T\mathbf{D}_k - \mathrm{vec}^{-1}(P_{\mathcal{B}_1^\eta}(\mathrm{vec}(\mathbf{D}_k^T\mathbf{D}_k - \mathbf{I}))). \tag{22}$$

The proximal mapping in (22) requires column projection onto the ℓ_1 norm-ball which can be efficiently performed using the method proposed in [32].

[2] We have removed the subscript of α_i for simplicity.

2.2. Updating $\mathbf{D}$

The dictionary update problem is

$$\min_{\mathbf{D}\in\mathcal{D}} \ \{f(\mathbf{D}) + r_d(\mathbf{D})\}, \tag{23}$$

where $f(\mathbf{D}) \triangleq F(\mathbf{D}, \mathbf{G}_{k+1})$. This problem is very similar to (6), and it does not have a closed-form solution in general. Alternatively, we use a linearized proximal gradient method [29] to solve it. This is achieved, by replacing f with its quadratic approximation around the previous estimate of $\mathbf{D}$. Doing so, the update problem is

$$\mathbf{D}^{k+1} = \underset{\mathbf{D}}{\operatorname{argmin}} \left\{ f(\mathbf{D}_k) + \nabla^T f(\mathbf{D}_k)(\mathbf{D} - \mathbf{D}_k) + \frac{1}{2\mu_d}\|\mathbf{D} - \mathbf{D}_k\|_F^2 + r_d(\mathbf{D}) \right\}, \tag{24}$$

in which

$$\nabla f(\mathbf{D}) = (\mathbf{D}\mathbf{X} - \mathbf{Y})\mathbf{X}^T + \frac{2}{\alpha}\mathbf{D}(\mathbf{D}^T\mathbf{D} - \mathbf{G}_{k+1}) \tag{25}$$

is the gradient of f, and $\mu_d > 0$. Problem (24) can be equivalently written as

$$\mathbf{D}^{k+1} = \underset{\mathbf{D}}{\operatorname{argmin}} \ \left\{ \frac{1}{2}\|\mathbf{D} - \tilde{\mathbf{D}}_k\|_F + r_d(\mathbf{D}) \right\} = P_{\mathcal{D}}(\tilde{\mathbf{D}}_k), \tag{26}$$

with $\tilde{\mathbf{D}}_k \triangleq \mathbf{D}_k - \mu_d \nabla f(\mathbf{D}_k)$. In short, using this approach the dictionary is updated by performing one-step gradient descent followed by a projection onto $\mathcal{D}$. The step-size parameter, μ_d, is determined by the Lipschitz constant of ∇f, denoted by L. It is first shown that ∇f is Lipschitz continuous. To do so, let us define

$$f_1(\mathbf{D}) \triangleq \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2, \ f_2(\mathbf{D}) \triangleq \frac{1}{2}\|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{k+1}\|_F^2. \tag{27}$$

So, $\nabla f(\mathbf{D}) = \nabla f_1(\mathbf{D}) + \frac{1}{\alpha}\nabla f_2(\mathbf{D})$. It is already known [1] that ∇f_1 is Lipschitz with constant $L_1 = \|\mathbf{X}^T\mathbf{X}\|$, where $\|.\|$ denotes the matrix spectral norm, defined as the maximum singular value. For ∇f_2, we have the following lemma:

Lemma 2. *The gradient of f_2, as defined in* (27)*, is Lipschitz continuous over $\mathcal{D}$. That is, there exists a constant $L_2 > 0$ such that for all $\mathbf{D}_1, \mathbf{D}_2 \in \mathcal{D}$*

$$\|\nabla f_2(\mathbf{D}_2) - \nabla f_2(\mathbf{D}_1)\|_F \leq L_2\|\mathbf{D}_2 - \mathbf{D}_1\|_F. \tag{28}$$

Proof: See Appendix B.

The final alternating minimization approach for solving (17) consists of iterating between (22) and (26). The following theorem based on [33] establishes the convergence of the generated sequence.

Theorem 3. *Let* $\{\mathbf{D}_k, \mathbf{G}_k\}$ *be the sequence generated by* (26) *and* (22). *Assume further that* $\mu_d \in (0, 1/L]$. *Then, any accumulation point of* $\{\mathbf{D}_k, \mathbf{G}_k\}$ *converges to a critical point of* $H(\mathbf{D}, \mathbf{G})$ *defined in* (17).

Proof: See Appendix C.

3. Proposed constrained algorithm

Our next algorithm, called CINC-DL, for constrained incoherent DL, aims at directly solving (9). As already discussed, this is in contrast to the approach of IPR-DL which first updates the dictionary neglecting the MC constraint and then iteratively optimizes the result to satisfy the MC constraint. By replacing $\mu(\mathbf{D})$ with its equivalent definition in (13), problem (9) becomes

$$\min_{\mathbf{D}\in\mathcal{D}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{D}^T\mathbf{D}-\mathbf{I}\|_\infty \leq \mu_0. \tag{29}$$

Our strategy for solving the above problem is the same as the one used in RINC-DL. To this end, as before, let us define the auxiliary variable $\mathbf{G} \triangleq \mathbf{D}^T\mathbf{D}$. Problem (29) is then equivalent to

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{G}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \ \text{ s.t. } \ \|\mathbf{G}-\mathbf{I}\|_\infty \leq \mu_0, \ \mathbf{G}=\mathbf{D}^T\mathbf{D}. \tag{30}$$

Using the penalty method, the final problem to be solved is the same as (17) with the difference that, here, $r_g(\mathbf{G}) \triangleq \mathcal{I}_{\hat{\mathcal{G}}_{\mu_0}}$, where

$$\hat{\mathcal{G}}_{\mu_0} \triangleq \{\mathbf{G} \mid \ \|\mathbf{G}-\mathbf{I}\|_\infty \leq \mu_0\}. \tag{31}$$

Now, let us focus on the update problem of $\mathbf{G}$, which can be expressed as

$$\mathbf{G}_{k+1} = \mathbf{I} + \underset{\mathbf{G}\in\mathcal{B}_\infty^{\mu_0}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{G}-\tilde{\mathbf{G}}_k\|_F^2, \tag{32}$$

where $\tilde{\mathbf{G}}_k \triangleq \mathbf{D}_k^T\mathbf{D}_k - \mathbf{I}$. The second term in the right-hand side of (32) is clearly the projection of $\tilde{\mathbf{G}}_k$ onto the ℓ_∞ matrix norm-ball of radius μ_0. The projection is unique, because the constraint set $\mathcal{B}_\infty^{\mu_0}$ is convex. To perform this projection, we use the following lemma:

Lemma 4. *The projection of a matrix $\mathbf{U}^0$ onto the ℓ_∞ matrix norm-ball $\mathcal{B}_\infty^r$, denoted by $\mathbf{U}^p \triangleq P_{\mathcal{B}_\infty^r}(\mathbf{U}^0)$, is characterized by*

$$u_{ij}^p = \begin{cases} sgn(u_{ij}^0)\cdot r & |u_{ij}^0| > r \\ u_{ij}^0 & otherwise \end{cases} \tag{33}$$

Proof: See Appendix D.

Therefore, the final update formula for $\mathbf{G}_{k+1}$ is

$$\mathbf{G}_{k+1} = \mathbf{I} + P_{\mathcal{B}_\infty^{\mu_0}}(\mathbf{D}_k^T\mathbf{D}_k - \mathbf{I})\cdot \tag{34}$$

The update problem for $\mathbf{D}_{k+1}$ is the same as (23). So, we omit the details. Iterating between (34) and (26) yields approximate solutions to (30). Convergence of the generated sequence is guaranteed by the following theorem, whose proof is very similar to that of Theorem 1.

Theorem 5. *Let $\{\mathbf{D}_k, \mathbf{G}_k\}$ be the sequence generated by* (26) *and* (34)*. Assume further that $\mu_d \in (0, 1/L]$. Then, any accumulation point of $\{\mathbf{D}_k, \mathbf{G}_k\}$ converges to a critical point of $H(\mathbf{D}, \mathbf{G})$ defined in* (17)*.*

A detailed description of the proposed algorithms is given in Algorithm 1. In this algorithm, $\mathrm{SA}(\mathbf{Y}, \mathbf{D}, \tau)$ stands for the sparse representation matrix of $\mathbf{Y}$ over $\mathbf{D}$ obtained via a sparse approximation algorithm with parameter τ. This parameter may be the maximum allowed number of atoms to be used in representations, an upper-bound on approximation error, or regularization parameter of sparsity promoting function. Note that the main difference between RINC-DL and CINC-DL lies in the $\mathbf{G}$-update step, i.e., line 12 of Algorithm 1.

Algorithm 1 Proposed algorithms (RINC-DL and CINC-DL)

Require: $\mathbf{Y}$, $\mathbf{D}_0$, μ_0 or λ, τ, c, L_2, ϵ, I, J
Initialization: $\mathbf{D} = \mathbf{D}_0$, $\mathbf{G} = \mathbf{0}$
while stopping criterion for DL not met **do**
    1. Sparse approximation: $\mathbf{X} = \mathrm{SA}(\mathbf{Y}, \mathbf{D}, \tau)$
    2. Dictionary update:
    $L_1 = \|\mathbf{X}^T\mathbf{X}\|$
    $\alpha = 3 \cdot \|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_\infty$
    $i = 1$
    while $i \leq I$ and $\|\mathbf{G} - \mathbf{D}^T\mathbf{D}\|_F > \epsilon$ **do**
        $\mu_d = 1/(L_1 + \alpha^{-1}L_2)$
        for $j = 1, 2, \cdots, J$ **do**
            $$\begin{cases} \mathbf{G} & = \mathbf{D}_k^T\mathbf{D} - \mathrm{vec}^{-1}(P_{\mathcal{B}_1^\eta}(\mathrm{vec}(\mathbf{D}^T\mathbf{D} - \mathbf{I}))) & \text{(RINC-DL)} \\ \mathbf{G} & = \mathbf{I} + P_{\mathcal{B}_\infty^{\mu_0}}(\mathbf{D}^T\mathbf{D} - \mathbf{I}) & \text{(CINC-DL)} \end{cases}$$
            $\mathbf{D} = P_{\mathcal{D}}(\mathbf{D} - \mu_d \nabla f(\mathbf{D}))$
        end for
        $\alpha_{i+1} = c \cdot \alpha_i$
        $i \leftarrow i + 1$
    end while
end while
Output: $\mathbf{D}$, $\mathbf{X}$

4. Discussion

Similar to IPR-DL, CINC-DL is designed to be used in situations where a fixed upper-bound on $\mu(\mathbf{D})$ is desired. On the other hand, RINC-DL targets applications in which the goal is to make a certain trade-off between minimizing the approximation error and $\mu(\mathbf{D})$. Compared with CINC-DL and IPR-DL, RINC-DL can be reduced to a plain DL, *i.e.*, without the MC constraint, by simply setting $\lambda = 0$. An advantage of CINC-DL (and also RINC-DL) over IPR-DL is that, by directly solving the constrained low-coherence DL problem in an

iterative fashion, CINC-DL benefits from *warm-start* during the DL iterations. That is, the previous estimate of $\mathbf{D}$ is used as the initialization point for the next DL iteration. In this way, the MC values are gradually decreased along DL iterations. In contrast, due to the non-iterative nature of IPR-DL, the dictionary has to be optimized in every DL iteration to satisfy the target level of MC.

In what follows, we compare the iteration complexity of the proposed algorithms and that of IPR-DL. The complexity is evaluated according to the number of required floating-point operations (flops). Note that since the SA stage is common to all the algorithms, only the DU stage is considered.

As discussed in [25], the iteration complexity of IPR-DL is dominated by computation of the EVD of the Gram matrix $\mathbf{D}^T\mathbf{D}$ which requires $\mathcal{O}(N^3)$ operations, computation of the covariance matrix $\mathbf{C} = (\mathbf{DX})\mathbf{Y}^T$ needing $\mathcal{O}(n^2M)$ operations, and computation of the SVD of the covariance matrix which costs $\mathcal{O}(n^3)$ operations. For our proposed algorithms, the iteration complexity is determined by the $\mathbf{G}$-update stages and the inner DU loops. Every column projection onto the ℓ_1 norm-ball costs $\mathcal{O}(N)$ operations according to [32]. So, the full projection performed in line 12 of the RINC-DL algorithm costs $\mathcal{O}(N^2)$ operations. The $\mathbf{G}$-update for the CINC-DL algorithm is a simple thresholding which costs $\mathcal{O}(N^2)$ operations. The inner DU stage for the two algorithms has a computational complexity of $\mathcal{O}(nNM + nN^2)$. The full investigation of the iteration complexity of the algorithms, neglecting constant numbers and after some simplifications, are outlined in Table 1. From this table, it is concluded that for all the algorithms the computational complexity is linear in M, the number of training signals. Also, CINC-DL and RINC-DL have the same order of computations. More importantly, the computational complexity of IPR-DL has a cubic growth in both n (signal dimension) and N (number of atoms), while it is linear in n and square in N for our proposed algorithms. This makes our proposed algorithms favorable for high-dimensional data settings.

Table 1: Iteration complexity of the algorithms.

Algorithm	Complexity
IPR-DL	$\mathcal{O}(nNM + MN^2 + 3n^2N + 2n^3 + 2N^3)$
CINC-DL	$\mathcal{O}(nNM + nN^2)$
RINC-DL	$\mathcal{O}(nNM + nN^2)$

5. Simulation results

The promising performance of low-coherence DL algorithms has already been verified in many applications, including face recognition, object classification, image denoising [23], functional magnetic resonance imaging (fMRI) [22], and sparse approximation of musical signals [24, 25]. In this section, we are going to compare the performance of our proposed algorithms with those of existing low-coherence DL algorithms, to see how well they can make a balance between reducing representation error and the mutual coherence of dictionary. To this end, we did an experiment on learning low mutual coherence dictionaries for natural image blocks. Furthermore, as another experiment, we considered the problem of discriminative dictionary learning for image classification. From the first group, the algorithm proposed in [21] was chosen, which we call bounded self-coherence DL (BSC-DL). BSC-DL uses the limited-memory BFGS (l-BFGS) algorithm [34] to solve (6). From the constrained low-coherence DL algorithms, IPR-DL was chosen, which is the state-of-the-art. To have a rough measure of the computational loads of the algorithms, their runtimes are reported. Our simulations were carried out in MATLAB environment on a 64 bit Windows 7 operating system with 8 GB RAM and an Intel core i7 CPU.

The rest of this section is organized as follows. Section 5.1 presents the experiments and results on image patch representation. The effects of the parameters of our proposed algorithms are studied in Subsection 5.1.2. Subsections 5.1.3 and 5.1.4 evaluate and compare the performance of our proposed algorithms with those of BSC-DL and IPR-DL. Finally, the experimental results on im-

age classification based on discriminative over-complete dictionary learning are discussed in Section 5.2.

5.1. Image patches representation

5.1.1. Setup

We consider learning low-coherence dictionaries for natural image patches. A number of $M = 50,000$ blocks of size 8×8 were randomly extracted from some benchmark images. The blocks were then converted to equivalent column vectors of length 64. We subtracted the mean from all vectors, and then normalized them. The resulting set of vectors were then used to form the training data matrix $\mathbf{Y}$. For all the competing algorithms, the initial dictionary was set as a DCT matrix of size 64×256, and OMP was used to implement the SA stages. In addition, the maximum number of participating atoms in the representations of all training signals was set to $s = 10$ ($\tau = 10$ in Algorithm 1). All the algorithms were run for 300 iterations (i.e., the number of iterations between sparse approximation and dictionary update). To measure the quality of the learned dictionaries, mean square error (MSE) was used which is computed as $\|\mathbf{Y} - \mathbf{DX}\|_F/(n \cdot M)$.

The parameters of the algorithms were set as follows. For the l-BFGS algorithm used in BSC-DL the parameters suggested by the authors of [21] were used. For IPR-DL, the maximum number of iterations in its decorrelation step was set to 100. For our proposed algorithms, outlined in Algorithm 1, we set $I = 100$, $J = 3$, $L_2 = 20$, $c = 0.85$, and $\epsilon = 0.05$, which resulted in promising performances. The effects of these parameters are explored in the next subsection.

5.1.2. Effects of parameters

In this subsection, the effects of the parameters of our proposed algorithms (*i.e.*, c, J, ϵ, and L_2) on their performances are experimentally demonstrated. To test the effect of each parameter, the others have been set to their default values, stated in the previous subsection. Moreover, the incoherence regularization

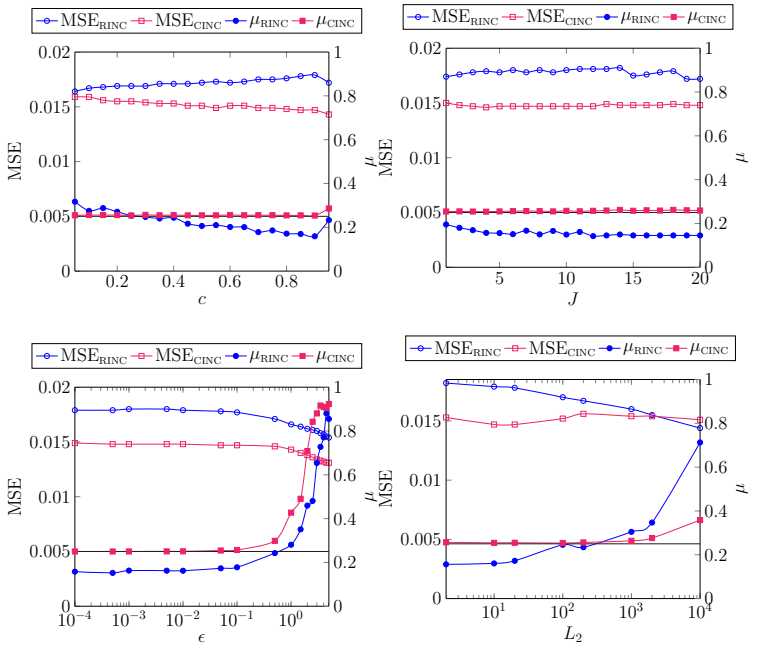


Figure 1: Plots of the final approximation errors and mutual coherences achieved by RINC-DL and CINC-DL versus different values of their parameters. The default values for the parameters are $c = 0.85$, $J = 3$, $\epsilon = 0.05$, and $L_2 = 20$. To test the effect of each parameter, the others were set to their default values.

parameter λ in RINC-DL, and the target MC in CINC-DL were set to $\lambda = 50$ and $\mu_0 = 0.25$, respectively. For all the parameters, final MSEs together with the mutual coherences of the learned dictionaries are plotted in Fig. 1, while Fig. 2 shows the corresponding runtimes. With these results in mind, the following conclusions are inferred concerning the role of each parameter:

- c: As demonstrated in Fig. 1 (a), the resulting MSE of CINC-DL decreases with increasing c, while its final MC remains approximately fixed at the target value. Moreover, the mutual coherence of the learned dictionary by RINC-DL decreases as c increases up to $c = 0.95$. On the other hand, larger values for c result in decreasing the convergence speeds of the algorithms, as confirmed by Fig. 2 (a).

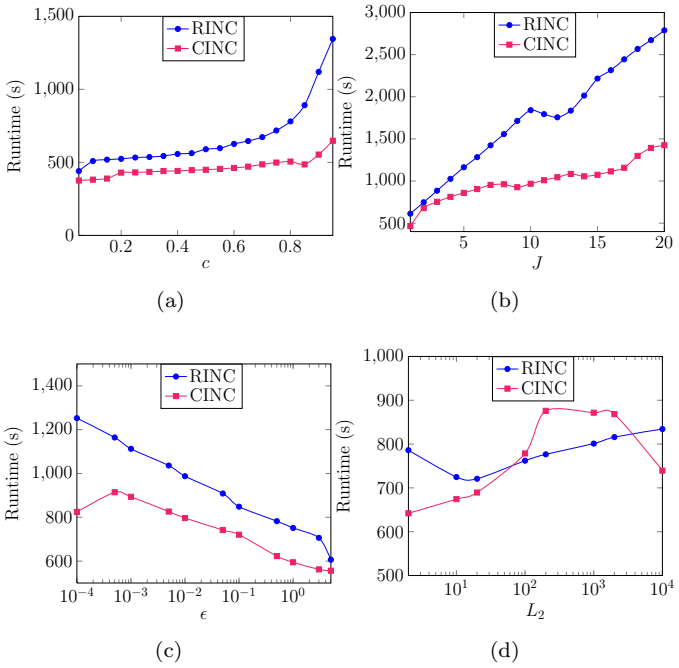
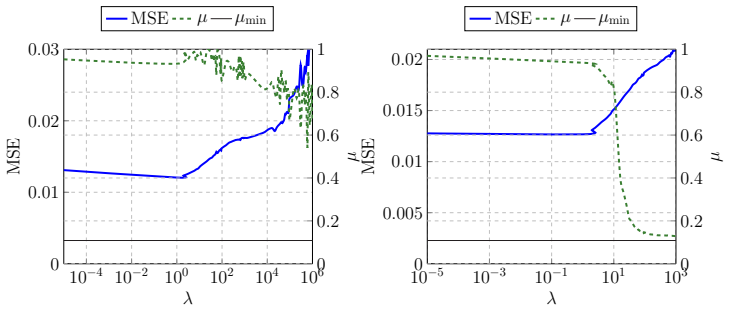


Figure 2: Runtimes of RINC-DL and CINC-DL versus different values of their parameters. The setting is the same as Fig. 1.

- J: Due to the inherent warm-starting in our proposed algorithms, a few iterations are sufficient for updating $\mathbf{G}$ and $\mathbf{D}$ in lines 11–14 of Algorithm 1. This is illustrated by Figs. 1 (b) and 2 (b). As clearly demonstrated, increasing J does not improve the performance of the algorithms too much.

- ϵ: As shown in Figs. 1 (c) and 2 (c), smaller values for the stopping tolerance, ϵ, lead to better performance in our proposed algorithms. This, however, increases the runtimes.

- L_2: Figures 1 (d) and 2 (d) show the performance of our algorithms for different values of L_2. It is observed that both of the algorithms have good performances for $L_2 \simeq 20$.



(a) BSC-DL (b) RINC-DL

Figure 3: Final approximation errors and the mutual coherences of the learned dictionaries achieved by BSC-DL and RINC-DL as functions of the regularization parameter, λ. The Welch bound is also depicted as μ_{min}.

5.1.3. Unconstrained algorithms

The performance of BSC-DL and RINC-DL, as two unconstrained algorithms, are compared in this subsection to see how well they make a trade-off between minimizing the mutual coherences of the learned dictionaries, and adapting the atoms to the training signals by reducing the associated approximation errors.

Figure 3 depicts plots of the final MCs and MSEs versus different values of the regularization parameters of the algorithms. Considering these results, it can be concluded that:

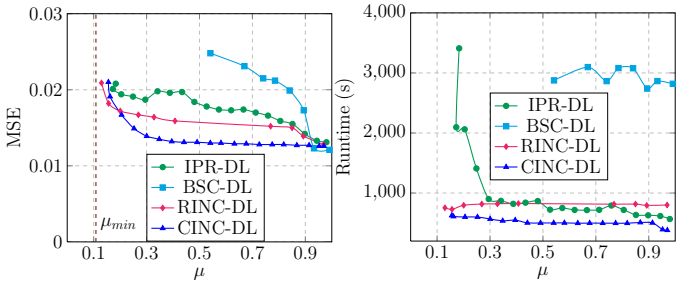
- As expected, for both algorithms increasing the incoherence regularization parameter, λ, results in a dictionary with low MC but with a worse approximation error (except for relatively small values of λ, which result in large MCs). In addition, while BSC-DL exhibits a highly variable behavior as λ increases, with its MC curve oscillating, RINC-DL has a more clear and monotonic trend. The oscillating behavior of BSC-DL might be due to the fact that this algorithm does not minimize the mutual coherence directly, as explained in Subsection 1.4. Moreover, the approximation errors reached by RINC-DL are considerably lower than those of BSC-DL.

- The minimum MC achieved by BSC-DL is about 0.54. On the other hand, RINC-DL achieves a minimum MC of 0.1296, which is quite close to $\mu_{\min} = 0.1085$. This further confirms the inefficiency of the existing regularization term, $\mathcal{R}(\mathbf{D})$, in imposing incoherency.

5.1.4. Constrained algorithms

In this subsection, the ability of the constrained algorithms, CINC-DL and IPR-DL, to upper-bound the MC while adapting the dictionary to the training data is evaluated.

The final approximation errors versus different mutual coherence levels for all the algorithms, including also BSC-DL and RINC-DL, are depicted in Fig. 4 (a). The associated runtimes are illustrated in Fig. 4 (b). By investigating the results shown in these figures, several conclusions can be reached. First, as can be noted, CINC-DL has the best overall performance, in terms of both approximation error (MSE) and runtime. However, RINC-DL shows a better performance in learning incoherent dictionaries with MCs near the WB. IPR-DL, on the other hand, has inferior performance to both CINC-DL and RINC-DL. Its runtime is also dependent on the mutual coherence level, in the way that it takes longer to learn dictionaries with MCs near the WB. This is due to the iterative projection that is used in the decorrelation step of IPR-DL. Projection onto matrices with very low-mutual coherences takes a lot of iterations. Finally, BSC-DL exhibits the worst performance, both in approximation errors and runtimes. Furthermore, the minimum mutual coherence it can reach (0.54) does not compare well with those of the other algorithms. Nevertheless, its approximation errors are slightly better than the ones achieved by the other algorithms for MCs larger than 0.9. Indeed, for these values of MCs, all the algorithms behave roughly as plain DL algorithms, without any MC penalty, and since BSC-DL uses a more advanced dictionary update algorithm, it slightly outperforms the others.



(a) MSE (b) Runtime

Figure 4: Comparison of the all competing algorithms, in terms of: (a) the final approximation errors, and (b) the runtimes, versus the corresponding mutual coherences of the learned dictionaries.

5.2. Image classification

In this section, we consider a face recognition task by learning a discriminative over-complete dictionary on some labeled training data. To this end, we follow the approach proposed in [35], consisting of solving the following problem:

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X},\mathbf{W}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2+\gamma\|\mathbf{H}-\mathbf{W}\mathbf{X}\|_F^2+\alpha\|\mathbf{W}\|_F^2, \tag{35}$$

subject to the constraint that each column of $\mathbf{X}$ contains at most T non-zeros entries. Here, $\mathbf{Y}\in\mathbb{R}^{n\times M}$ is the matrix of labeled training data, belonging to C different classes, $\mathbf{H}\in\mathbb{R}^{C\times M}$ is the label matrix, $\mathbf{W}\in\mathbb{R}^{C\times N}$ is a linear classifier, and γ and α are some trade-off parameters. This way, a linear classifier on the sparse coefficients is jointly trained with a sparsifying dictionary. An efficient algorithm, called discriminative K-SVD (DK-SVD), is proposed in [35] which extends the K-SVD algorithm to solve (35).

To learn a low-mutual coherence discriminative dictionary, we modify (35) by including the constraint $\mu(\mathbf{D})\leq\mu$. To solve the resulting problem, we follow a simple alternating minimization approach over all the variables, namely $\mathbf{D}$, $\mathbf{X}$, and $\mathbf{W}$. This is similar to a standard DL problem except for the additional step of updating $\mathbf{W}$. To update $\mathbf{X}$, we used the OMP algorithm, which is also used by DK-SVD. To update the dictionary with the mutual coherence constraint,

we used IPR-DL and CINC-DL. Assuming that $\mathbf{D}$ and $\mathbf{X}$ have already been updated, to update $\mathbf{W}$ the following problem needs to be solved:

$$\min_{\mathbf{W}} \ \gamma\|\mathbf{H} - \mathbf{W}\mathbf{X}\|_F^2 + \alpha\|\mathbf{W}\|_F^2 \tag{36}$$

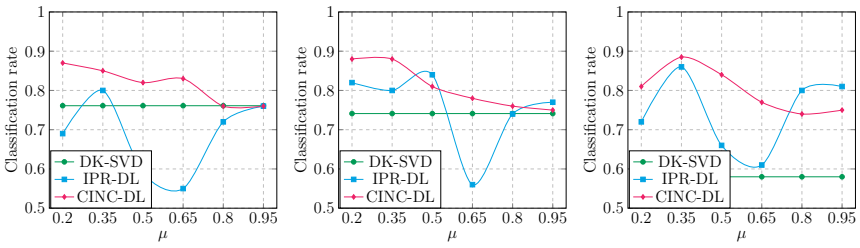
which admits a closed-form solution given by:

$$\mathbf{W} = (\mathbf{H}\mathbf{X}^T)(\mathbf{X}\mathbf{X}^T + \frac{\alpha}{\gamma}\mathbf{I})^{-1}. \tag{37}$$

At test time, once the dictionary and linear classifier are learned, the sparse coefficients vector of a test sample is first computed over the trained dictionary and then passed to the linear classifier [35].

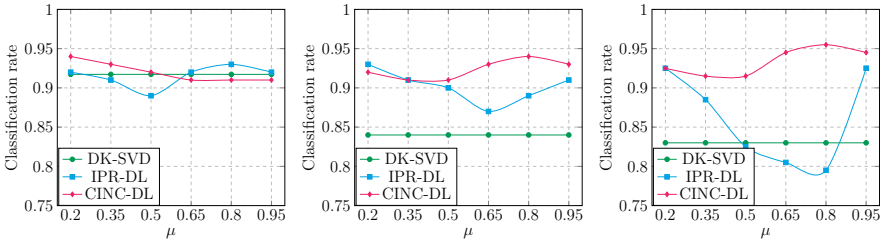
For this experiment, we considered two well-known face datasets, i.e., Extended YaleB and AR [35]. The Extended YaleB dataset contains about 2414 frontal face images of 38 individuals (1216 for training and 1198 for testing), each of size 192×168. Similar to [35], each face image was transformed to a feature vector of length 504 using a random projection. The AR dataset contains over 4000 face images of 126 people, from which we chose 2600 images, as done in [35]. This set was then randomly divided into a training set (2000 images) and test set (600 images). Similar to the Extended YaleB dataset, a random feature vector of length 540 was then obtained for each face image. The total number of iterations for all the competing algorithms, i.e., DK-SVD, IPR-DL, and CINC-DL, was set to 50, which was enough for their convergence. For the sparsity level, T, we chose $T = 16$ for Extended YaleB and $T = 10$ for AR, as suggested in [35]. We also set $\alpha = \gamma = 1$ in (35), following [35]. The parameters of CINC-DL were set as $c = 0.25$, $J = 1$, $I = 100$, and $\epsilon = 0.005$. We experimented with different number of atoms, i.e., $N = 600, 800, 1000$. To initialize $\mathbf{D}$, K-SVD was run over the training data, and then $\mathbf{W}$ was initialized using (37). To alleviate the effect of initialization, each experiment was repeated five times and the average results were reported.

The classification rates versus mutual coherence for different number of atoms are shown in Figs. 5 and 6. The mutual coherence of the learned dictionary by DK-SVD in all the cases is around 0.98. By investigating these figures



(a) $N = 600$ (b) $N = 800$ (c) $N = 1000$

Figure 5: Classification rates versus mutual coherence of the learned dictionary on the **AR dataset**. Each figure corresponds to a specific number of atoms N.



(a) $N = 600$ (b) $N = 800$ (c) $N = 1000$

Figure 6: Classification rates versus mutual coherence of the learned dictionary on the **Extended YaleB dataset**. Each figure corresponds to a specific number of atoms N.

we can see a clear advantage of constraining the mutual coherence of the dictionary exhibited in the results of IPR-DL and CINC-DL. A consistent behavior shown by DK-SVD is that the classification rate decreases by increasing the number of atoms. This may be explained by the fact that as the number of atoms increases, so does the number of trainable parameters, and with a fixed number of training samples, the chance of over-fitting increases. This effect is not so significant in the case of IPR-DL and CINC-DL, as the mutual coherent constraint acts as a regularizer, thus avoiding over-fitting. Considering higher dimensions (larger dictionary size) is also beneficial in the sense that the resulting sparse coefficients vectors become more discriminative. The performances of IPR-DL and CINC-DL on the AR dataset are much better than that of DK-

SVD. It is also noticeable that CINC-DL shows a much more consistent behavior than IPR-DL across mutual coherence and number of atoms. In particular, in almost all the cases, especially for AR, the best performance of CINC-DL occurs in small values of the mutual coherence, and it consistently outperforms IPR-DL.

It should be noted that our algorithmic framework can be applied to other variants of DK-SVD, e.g. [36] as well, by adding a mutual coherence constraint to the dictionary update step. The LC-KSVD algorithm proposed in [36], however, has been proven to be equivalent to DK-SVD up to a proper choice of regularization parameters [37]. In our experiments, we also observed no considerable difference between the results of DK-SVD and LC-KSVD.

6. Conclusion and future work

In this paper, we addressed dictionary learning (DL) with mutual coherence constraint. In this regard, we proposed an unconstrained algorithm which targets the regularized DL problem in which the mutual coherence function is used as the regularizer. A new constrained algorithm was also proposed which solves the DL problem under a mutual coherence constraint. Both algorithms are based on penalty methods and proximal approaches. Our computational analysis revealed that, compared with the state-of-the-art algorithm of [25], our new algorithms are more favorable for high-dimensional applications. In addition, our experimental results on learning low-coherence dictionaries for natural image patches as well as image classification based on discriminative over-complete dictionary learning confirmed the superiority of our new algorithms over the previous ones.

One interesting future research direction would be to investigate the application of the proposed algorithms in learning discriminative sub-dictionaries for classification tasks. As proposed in [38], for classification using dictionary learning, a different dictionary is learned for each data class in such a way that the dictionaries for different classes are as incoherent with each other as possible.

This offers a different approach than the one discussed in Section 5.2. More precisely, assuming that the training data are arranged in $\mathbf{Y}$ as $\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_C]$, where $\mathbf{Y}_i$ denotes training data belonging to class i, the following DL problem is proposed in [38]:

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}} \ \frac{1}{2}\sum_{i=1}^{C}\left\{\|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 + \lambda\|\mathbf{X}_i\|_1\right\} + \eta\sum_{i\neq j}\|\mathbf{D}_i^T\mathbf{D}_j\|_F^2, \tag{38}$$

where $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_C]$, $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_C]$ and $\mathbf{D}_i \in \mathbb{R}^{n\times N_i}$ is a dictionary with N_i atoms associated with the i-th class. The last term in (38) is responsible for decreasing the mutual coherence between any two distinct dictionaries. This term, however, minimizes the *average* mutual coherence. With our proposed idea, the following problem should be solved:

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}} \ \frac{1}{2}\sum_{i=1}^{C}\left\{\|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 + \lambda\|\mathbf{X}_i\|_1\right\} + \eta\cdot\max_{i\neq j}\|\mathbf{D}_i^T\mathbf{D}_j\|_F^2 \tag{39}$$

In this way, every two distinct dictionaries would be learned such that they are as incoherent as possible and also matched to the training signals of their own class. Designing an efficient algorithm to solve the above problem can be pursued as a future work.

Another idea to pursue as a future work is to use a Bregman iteration technique [39] to handle the equality constraints in (16) and (30). The advantage of this technique over the penalty method is that instead of a decreasing sequence for the penalty parameter, a fixed value is used. This would avoid the numerical instability that occurs when the penalty parameter goes to zero [39]. Another advantage would be a faster convergence rate, thus reducing the computational complexity of the overall algorithm [39].

Appendix A. Proof of Lemma 1

Consider the definition of $\text{prox}_g(.)$ given below

$$\text{prox}_g(\mathbf{U}) = \underset{\mathbf{Z}}{\text{argmin}} \ \frac{1}{2}\|\mathbf{Z} - \mathbf{U}\|_F^2 + \eta\|\mathbf{Z}\|_\infty. \tag{A.1}$$

Defining $\mathbf{u} \triangleq \text{vec}(\mathbf{U})$ and $\mathbf{z} \triangleq \text{vec}(\mathbf{Z})$, it is straightforward to show that problem (A.1) can be expressed in the following vectorized form

$$\text{vec}(\text{prox}_g(\mathbf{U})) = \underset{\mathbf{z}}{\text{argmin}} \ \frac{1}{2}\|\mathbf{z}-\mathbf{u}\|_2^2 + \eta\|\mathbf{z}\|_\infty. \tag{A.2}$$

We then use the Moreau decomposition [29] to derive the proximal mapping of the ℓ_∞ norm. The Moreau decomposition states that the following relation holds for any convex function f

$$\mathbf{x} = \text{prox}_f(\mathbf{x}) + \text{prox}_{f^*}(\mathbf{x}), \tag{A.3}$$

where

$$f^*(\mathbf{y}) \triangleq \sup_{\mathbf{x}} \{\langle \mathbf{y}, \mathbf{x}\rangle - f(\mathbf{x})\} \tag{A.4}$$

is the convex conjugate of f [40]. For ηf, using (A.4) it can be verified that the following generalization of (A.3) holds

$$\mathbf{x} = \text{prox}_{\eta f}(\mathbf{x}) + \eta\text{prox}_{f^*/\eta}(\mathbf{x}/\eta). \tag{A.5}$$

For our problem, let $f = \|.\|_\infty$. The convex conjugate of f is $f^* = \mathcal{I}_{\mathcal{B}_1^1}$, that is, the indicator function of the unit ℓ_1 norm-ball [40]. Then, using (A.3) and (12), it is verified that

$$\text{prox}_f(\mathbf{x}) = \mathbf{x} - P_{\mathcal{B}_1^1}(\mathbf{x}). \tag{A.6}$$

Finally, using (A.5) we have $\text{prox}_{\eta f}(\mathbf{x}) = \mathbf{x} - P_{\mathcal{B}_1^\eta}(\mathbf{x})$, which by using vec(.) and $\text{vec}^{-1}(.)$, the result in (20) is obtained.

Appendix B. Proof of Lemma 2

The proof is mainly based on the submultiplicativity property of the Frobenius norm [41] which states that for any two matrices $\mathbf{A}$ and $\mathbf{B}$ of consistent dimensions the following inequality holds

$$\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F \cdot \|\mathbf{B}\|_F. \tag{B.1}$$

To begin the proof, first recall from (25) that the gradient of f_2 is given by $\nabla f_2(\mathbf{D}) = 2\mathbf{D}(\mathbf{D}^T\mathbf{D} - \mathbf{G}_{k+1})$. It is then straightforward to verify the following

equalities

$$\nabla f_2(\mathbf{D}_1) - \nabla f_2(\mathbf{D}_2) = 2\Big(\mathbf{D}_1\mathbf{D}_1^T\mathbf{D}_1 - \mathbf{D}_1\mathbf{G}_{k+1} - \mathbf{D}_2\mathbf{D}_2^T\mathbf{D}_2 + \mathbf{D}_2\mathbf{G}_{k+1}\Big) = 2\Big((\mathbf{D}_1 - \mathbf{D}_2)\mathbf{D}_1^T\mathbf{D}_1 + \mathbf{D}_2(\mathbf{D}_1 - \mathbf{D}_2)^T\mathbf{D}_1 + \mathbf{D}_2\mathbf{D}_2^T(\mathbf{D}_1 - \mathbf{D}_2) - (\mathbf{D}_1 - \mathbf{D}_2)\mathbf{G}_{k+1}\Big)\cdot \quad \text{(B.2)}$$

Then, using the triangle inequality for matrix norms along with the application of (B.1) results in

$$\|\nabla f_2(\mathbf{D}_1) - \nabla f_2(\mathbf{D}_2)\|_F \leq 2\Big(\|\mathbf{D}_1\|_F^2 + \|\mathbf{D}_1\|_F\|\mathbf{D}_2\|_F + \|\mathbf{D}_2\|_F^2 + \|\mathbf{G}_{k+1}\|_F\Big)\cdot\|\mathbf{D}_1 - \mathbf{D}_2\|_F \quad \text{(B.3)}$$

On the other hand, it is easy to show that

$$\forall \mathbf{D} \in \mathcal{D}:\ \|\mathbf{D}\|_F = \sqrt{N}, \quad \text{(B.4)}$$

Finally, from (B.3) and (B.4) the inequality in (28) is readily followed with

$$L_2 = 6N + 2\|\mathbf{G}_{k+1}\|_F\cdot \quad \text{(B.5)}$$

Appendix C. Proof of Theorem 3

To prove Theorem 3, we use existing results, especially the convergence proof provided in [33]. Before proceeding, notice the following necessary definitions and useful lemmas:

Definition 3 (Subdifferential [42])**.** *The subdifferential of a proper, lower semi-continuous function g at $\mathbf{x} \in \mathbb{R}^n$ is defined as*

$$\partial g(\mathbf{x}) \triangleq \Big\{\zeta \in \mathbb{R}^n |\ \exists\ \mathbf{x}_k \to \mathbf{x}, g(\mathbf{x}_k) \to g(\mathbf{x}), \zeta_k \to \zeta, \zeta_k \in \hat{\partial} g(\mathbf{x}_k)\Big\}, \quad \text{(C.1)}$$

in which, $\hat{\partial} g(\mathbf{x})$ is the Fréchet subdifferential of g at $\mathbf{x} \in \mathbb{R}^n$ defined as

$$\hat{\partial} g(\mathbf{x}) \triangleq \Big\{\zeta \in \mathbb{R}^n |\ \liminf_{\mathbf{v}\to\mathbf{x}_{\mathbf{v}\neq\mathbf{x}}} \frac{1}{\|\mathbf{x}-\mathbf{v}\|_2^2}\cdot\Big(g(\mathbf{v}) - g(\mathbf{x}) - \langle \mathbf{v}-\mathbf{x}, \zeta\rangle\Big) \geqslant 0\Big\}. \quad \text{(C.2)}$$

It is said that $\mathbf{x}^*$ is a critical point of a proper, lower semi-continuous (PLSC) function f if $0 \in \partial f(\mathbf{x}^*)$ [42].

Lemma 6 ([33]). *Let $h = f + g$, where f is continuously differentiable and g is convex. Then, $\forall \mathbf{x} \in \mathrm{dom}h$*

$$\partial h(\mathbf{x}) = \nabla f(\mathbf{x}) + \partial g(\mathbf{x}).$$

Using subdifferential properties [42] and invoking the above lemma, for the subdifferential of $H(\mathbf{D}, \mathbf{G})$ defined in (17) we have

$$\partial H(\mathbf{D}, \mathbf{G}) = \Big(\partial_d H(\mathbf{D}, \mathbf{G}), \partial_g H(\mathbf{D}, \mathbf{G})\Big) = \Big(\nabla_d F(\mathbf{D}, \mathbf{G}) + \partial r_d(\mathbf{D}), \nabla_g F(\mathbf{D}, \mathbf{G}) + \partial r_g(\mathbf{G})\Big). \quad \text{(C.3)}$$

Proposition 1 ([33]). *Let $\{(\mathbf{x}_k, \mathbf{u}_k)\}_{k=0}^{\infty}$ be a sequence in Graph$(\partial g) \triangleq \{(\mathbf{z}, \mathbf{v}) \mid \mathbf{v} \in \partial g(\mathbf{z})\}$ that converges to $(\mathbf{x}, \mathbf{u})$ as $k \to \infty$. By the definition of ∂g, if $g(\mathbf{x}_k)$ converges to $g(\mathbf{x})$ as $k \to \infty$, then $(\mathbf{x}, \mathbf{u}) \in$ Graph(∂g).*

Lemma 7 (Descent lemma [43]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a C^1-smooth function with L-Lipschitz continuous gradient ∇f on $\mathrm{dom}f$. Then for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}f$ it holds that*

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|_2^2. \quad \text{(C.4)}$$

To prove Theorem 3, we borrow ideas from recent work on proximal algorithms for non-convex problems [44, 43, 33]. To this end, first note that our proposed algorithm performs the following iterations[3], repeatedly, to update $\mathbf{D}$ and $\mathbf{G}$:

$$\mathbf{G}_{k+1} = \underset{\mathbf{G}}{\mathrm{argmin}} \Big\{ F(\mathbf{D}_k, \mathbf{G}_k) + \nabla_g^T F(\mathbf{D}_k, \mathbf{G}_k)(\mathbf{G} - \mathbf{G}_k) + \frac{1}{2\mu_g}\|\mathbf{G} - \mathbf{G}_k\|_F^2 + r_g(\mathbf{G}) \Big\} \quad \text{(C.5)}$$

[3]Here, by an abuse of notation and for simplicity, we define $\mathbf{A}^T\mathbf{B}$ as the inner-product between the two matrices $\mathbf{A}$ and $\mathbf{B}$.

and

$$\mathbf{D}_{k+1} = \underset{\mathbf{D}}{\operatorname{argmin}} \Big\{ F(\mathbf{D}_k, \mathbf{G}_{k+1}) + \nabla_d^T F(\mathbf{D}_k, \mathbf{G}_{k+1})(\mathbf{D} - \mathbf{D}_k) + \frac{1}{2\mu_d}\|\mathbf{D} - \mathbf{D}_k\|_F^2 + r_d(\mathbf{D}) \Big\}. \quad \text{(C.6)}$$

Moreover, $\mu_g \in (0, 1/L_g]$ and $\mu_d \in (0, 1/L_d]$, with L_g and L_d being the Lipschitz constants of $\nabla_g F$ and $\nabla_d F$, respectively. It is straightforward to show that $L_g = 1/\alpha$. It was also shown in Lemma 2 that $\nabla_d F$ is Lipschitz, and a Lipschitz constant was derived.

To justify the equivalence of (C.5) and (19), note that the former can be written as

$$\mathbf{G}_{k+1} = \underset{\mathbf{G}}{\operatorname{argmin}} \ \frac{1}{2\mu_g}\|\mathbf{G} - \Big(\mathbf{G}_k - \mu_g \nabla_g F(\mathbf{D}_k, \mathbf{G}_k)\Big)\|_F^2 + r_g(\mathbf{G}). \quad \text{(C.7)}$$

Then, by letting $\mu_g \to 1/L_g = \alpha$ and

$$\nabla_g F(\mathbf{D}_k, \mathbf{G}_k) = \frac{1}{\alpha}(\mathbf{G}_k - \mathbf{D}_k^T \mathbf{D}_k) \quad \text{(C.8)}$$

together with a simple change of variable, it turns out that (C.7), and thus (C.5), is equivalent to (19).

From (C.5) it follows that

$$\nabla_g^T F(\mathbf{D}_k, \mathbf{G}_k)(\mathbf{G}_{k+1} - \mathbf{G}_k) + \frac{1}{2\mu_g}\|\mathbf{G}_{k+1} - \mathbf{G}_k\|_F^2 + r_g(\mathbf{G}_{k+1}) \le r_g(\mathbf{G}_k). \quad \text{(C.9)}$$

On the other hand, using Lemma 7 we have

$$F(\mathbf{D}_k, \mathbf{G}_{k+1}) \le F(\mathbf{D}_k, \mathbf{G}_k) + \nabla_g^T F(\mathbf{D}_k, \mathbf{G}_k)(\mathbf{G}_{k+1} - \mathbf{G}_k) + \frac{L_g}{2}\|\mathbf{G}_{k+1} - \mathbf{G}_k\|_F^2. \quad \text{(C.10)}$$

Adding both sides of (C.9) and (C.10) results in

$$F(\mathbf{D}_k, \mathbf{G}_k) + r_g(\mathbf{G}_k) \ge F(\mathbf{D}_k, \mathbf{G}_{k+1}) + r_g(\mathbf{G}_{k+1}) + (\frac{1}{2\mu_g} - \frac{L_g}{2})\|\mathbf{G}_{k+1} - \mathbf{G}_k\|_F^2. \quad \text{(C.11)}$$

Similarly,

$$F(\mathbf{D}_k, \mathbf{G}_{k+1}) + r_d(\mathbf{D}_k) \ge F(\mathbf{D}_{k+1}, \mathbf{G}_{k+1}) + r_d(\mathbf{D}_{k+1}) + (\frac{1}{2\mu_d} - \frac{L_d}{2})\|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2. \quad \text{(C.12)}$$

Adding both sides of (C.11) and (C.12), we obtain

$$H(\mathbf{D}_k, \mathbf{G}_k) - H(\mathbf{D}_{k+1}, \mathbf{G}_{k+1}) \geq \rho_g \|\mathbf{G}_{k+1} - \mathbf{G}_k\|_F^2 + \rho_d \|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2, \quad \text{(C.13)}$$

where, $\rho_g \triangleq (2/\mu_g - L_g/2) \geq 0$ and $\rho_d \triangleq (2/\mu_d - L_d/2) \geq 0$. The above inequality shows that the sequence of objective values $\{H(\mathbf{D}_k, \mathbf{G}_k)\}$ is non-increasing. Moreover, H is bounded from below. So, the whole sequence is convergent. Summing (C.13) for $k \geq 0$ results in

$$\sum_{k=0}^{\infty} \left\{ \rho_g \|\mathbf{G}_{k+1} - \mathbf{G}_k\|_F^2 + \rho_d \|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2 \right\} \leq H_0 - H_\infty, \quad \text{(C.14)}$$

where, $H_0 \triangleq H(\mathbf{D}_0, \mathbf{G}_0)$ and $H_\infty \triangleq H(\mathbf{D}_\infty, \mathbf{G}_\infty)$. Noting that $H_0 - H_\infty \geq 0$, the above inequality implies that $\mathbf{D}_{k+1} \to \mathbf{D}_k$ and $\mathbf{G}_{k+1} \to \mathbf{G}_k$. Moreover, since in each iteration, $\mathbf{D}$ is projected onto $\mathcal{D}$, which is a bounded set, so the sequence $\{\mathbf{D}_k\}_{k=0}^{\infty}$ is bounded. Noting the update formula of $\mathbf{G}$, this also results in boundedness of the $\mathbf{G}$ sequence. Therefore, the sequence $\{(\mathbf{D}_k, \mathbf{G}_k)\}_{k=0}^{\infty}$ is bounded. So, according to the Bolzano–Weierstrass theorem [45], there exists a convergent subsequence $\left\{(\mathbf{D}_{k_j}, \mathbf{G}_{k_j})\right\}_{j=0}^{\infty}$ that converges to an accumulation point, say $(\mathbf{D}^*, \mathbf{G}^*)$. We next prove that $(\mathbf{D}^*, \mathbf{G}^*)$ is a critical point of H. Since F is continuous and r_g and r_d are PLSC, we have $H(\mathbf{D}_{k_j}, \mathbf{G}_{k_j}) \to H(\mathbf{D}^*, \mathbf{G}^*)$ as $j \to \infty$. The optimality condition for (C.5) reads as

$$0 \in \nabla_g F(\mathbf{D}_k, \mathbf{G}_k) + \partial r_g(\mathbf{G}_{k+1}) + \frac{1}{\mu_g}(\mathbf{G}_{k+1} - \mathbf{G}_k). \quad \text{(C.15)}$$

Let us define

$$A_j^g \triangleq \nabla_g F(\mathbf{D}_{k_j+1}, \mathbf{G}_{k_j+1}) - \nabla_g F(\mathbf{D}_{k_j}, \mathbf{G}_{k_j}) - \frac{1}{\mu_g}(\mathbf{G}_{k_j+1} - \mathbf{G}_{k_j}). \quad \text{(C.16)}$$

Then, using (C.15) we can write

$$A_j^g \in \nabla_g F(\mathbf{D}_{k_j+1}, \mathbf{G}_{k_j+1}) + \partial r_g(\mathbf{G}_{k_j+1}) = \partial_g H(\mathbf{D}_{k_j+1}, \mathbf{G}_{k_j+1}). \quad \text{(C.17)}$$

Similar results can be obtained for $\mathbf{D}$. So, we have

$$(A_j^g, A_j^d) \in \partial H(\mathbf{D}_{k_j+1}, \mathbf{G}_{k_j+1}). \quad \text{(C.18)}$$

In addition, using (C.16) together with the Lipschitz continuity of $\nabla_g F$ we can write

$$\|A_j^g\|_F \leq (L_g + \frac{1}{\mu_g})\|\mathbf{G}_{k_j+1} - \mathbf{G}_{k_j}\|_F, \tag{C.19}$$

which by considering $\mathbf{G}_{k+1} \to \mathbf{G}_k$, results in $A_j^g \to \mathbf{0}$. For the $\mathbf{D}$-update a similar result holds. So, $(A_j^g, A_j^d) \to (\mathbf{0}, \mathbf{0})$, which together with $H(\mathbf{D}_{k_j}, \mathbf{G}_{k_j}) \to H(\mathbf{D}^*, \mathbf{G}^*)$, (C.18), and using Proposition 1 results in $(\mathbf{0}, \mathbf{0}) \in \partial H(\mathbf{D}^*, \mathbf{G}^*)$.

Appendix D. Proof of Lemma 4

First note that, by the definition of the ℓ_∞ norm, the following relation holds

$$\{\mathbf{U} \in \mathcal{B}_\infty^r\} \equiv \{\forall i, j : \ |u_{ij}| \leq r\}\cdot \tag{D.1}$$

Then, consider the definition of $P_{\mathcal{B}_\infty^r}(.)$

$$\mathbf{U}^p = \underset{\mathbf{U}\in\mathcal{B}_\infty^r}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{U} - \mathbf{U}^0\|_F^2, \tag{D.2}$$

which is equivalent to

$$\mathbf{U}^p = \underset{|u_{ij}|\leq r}{\operatorname{argmin}} \ \frac{1}{2}\sum_{i,j}(u_{ij} - u_{ij}^0)^2\cdot \tag{D.3}$$

The above problem is decomposable over u_{ij}'s, resulting in

$$\forall i, j : \ u_{ij}^p = \underset{|u|\leq r}{\operatorname{argmin}} \ \frac{1}{2}(u - u_{ij}^0)^2\cdot \tag{D.4}$$

But, this is a projection onto box constraints which is obtained by simply clipping u_{ij}^0's when their absolute values exceed r, and keeping them unchanged otherwise. This completes the proof.

Acknowledgement

This paper has been supported in part by Iran National Science Foundation (INSF) under contract no. 96000780, and by Research Deputy of Sharif University of Technology.

References

[1] M. Elad, Sparse and Redundant Representations, Springer, 2010.

[2] S. S. Chen, D. D. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (2001) 129–159.

[3] J. A. Tropp, S. J. Wright, Computational methods for sparse solution of linear inverse problems, Proceedings of the IEEE 98 (6) (2010) 948–958.

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

[5] S. Ravishankar, Y. Bresler, MR image reconstruction from highly undersampled k-space data by dictionary learning, IEEE Trans. Medical Imaging 30 (5) (2011) 1028–1041.

[6] S. Ravishankar, J. C. Ye, J. A. Fessler, Image reconstruction: From sparsity to data-adaptive methods and machine learning, IEEE: Special Issue on Biomedical Imaging & Analysis in the Age of Sparsity, Big Data and Deep Learning[Available online] arXiv preprint arXiv:1904.02816.

[7] S. Ravishankar, R. R. Nadakuditi, J. A. Fessler, Efficient sum of outer products dictionary learning (soup-dil) and its application to inverse problems, IEEE Transactions on Computational Imaging 3 (4) (2017) 694–709.

[8] B. Dumitrescu, P. Irofti, Dictionary learning algorithms and applications, Springer, 2018.

[9] E. J. Candès, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies, IEEE Trans. Info. Theory 52 (12) (2006) 5406–5425.

[10] S. Foucart, M.-J. Lai, Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$,, Appl. Comput. Harmonic Anal. 26 (3) (2009) 395–407.

[11] D. L. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Information Theory 47 (7) (2001) 2845–2862.

[12] T. Strohmer, H. R. W., Grassmannian frames with applications to coding and communication, Applied and Computational Harmonic Analysis 14 (3) (2003) 257–275.

[13] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressive Sensing, Applied and Numerical Harmonic Analysis, Birkhäuser Basel, 2013.

[14] A. M. Tillmann, M. E. Pfetsch, The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing, IEEE Trans. Inf. Theory 60 (2) (2014) 1248–1259.

[15] J. A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inform. Theory 50 (10) (2004) 2231–2242.

[16] R. Gribonval, P. Vandergheynst, On the exponential convergence of matching pursuit in quasi-incoherent dictionaries, IEEE Trans. Inf. Theory 52 (1) (2006) 255–261.

[17] J. A. Tropp, On the conditioning of random subdictionaries, Appl. Computat. Harmon. Anal. 25 (1) (2008) 1–24.

[18] R. Gribonval, K. Schnass, Dictionary identificationsparse matrix-factorization via ℓ_1 -minimization, IEEE Trans. Info. Theory 56 (7) (2010) 3523–3539.

[19] S. Wu, B. Yu, Local identifiability of l1-minimization dictionary learning: a sufficient and almost necessary condition, arXiv:1505.04363 (2015).
URL `arXiv:1505.04363`

[20] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, R. Tandon, Learning sparsely used overcomplete dictionaries via alternating minimization, arXiv:1310.7991v2 (2014).
URL `arXiv:1310.7991v2`

[21] C. D. Sigg, T. Dikk, J. M. Buhmann, Learning dictionaries with bounded self-coherence, IEEE Signal Proc. Letters 19 (12) (2012) 861–864.

[22] V. Abolghasemi, S. Ferdowsi, S. Sanei, Fast and incoherent dictionary learning algorithms with application to fMRI, Signal, Image and Video Processing (Springer) 9 (1) (2015) 147–158.

[23] C. Bao, Y. Quan, H. Ji, A convergent incoherent dictionary learning algorithm for sparse coding, in: Computer Vision – ECCV, Vol. 8694 of Lecture Notes in Computer Science, Springer, 2014, pp. 302–316.

[24] B. Mailhé, D. Barchiesi, M. D. Plumbley, INK-SVD: Learning incoherent dictionaries for sparse representations, in: IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2012, pp. 3573–3576.

[25] D. Barchiesi, M. D. Plumbley, Learning incoherent dictionaries for sparse approximation using iterative projections and rotations, IEEE Trans. on Signal Proc. 61 (8) (2013) 2055–2065.

[26] W. Chen, M. R. D. Rodrigues, I. J. Wassell, On the use of unitnorm tight frames to improve the average mse performance in compressive sensing applications, IEEE Signal Process. Lett. 19 (1) (2012) 8–11.

[27] B. Dumitrescu, Designing incoherent frames with only matrix-vector multiplications, IEEE Signal Processing Letters 24 (9) (2017) 1265–1269.

[28] G. Li, Z. Zhu, D. Yang, L. Chang, H. Bai, On projection matrix optimization for compressive sensing systems, IEEE Trans. on Signal Proc. 61 (11) (2013) 2887–2898.

[29] N. Parikh, S. Boyd, Proximal algorithms, Foundations and Trends in Optimization 1 (3) (2014) 123–231.

[30] M. Sadeghi, M. Babaie-Zadeh, C. Jutten, Regularized low-coherence overcomplete dictionary learning for sparse signal decomposition, in: Proceedings of 24th European Signal Processing Conference (EUSIPCO 2016), 2016.

[31] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, 1999.

[32] J. Duchi, S. Shalev-Shwartz, Y. Singer, T. Chandra, Efficient projections onto the l1-ball for learning in high dimensions, in: International Conference on Machine Learning (ICML), 2008.

[33] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1-2) (2014) 459–494.

[34] D. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Progr. 45 (1) (1989) 503–528.

[35] Q. Zhang, B. Li, Discriminative k-svd for dictionary learning in face recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[36] Z. Jiang, Z. Lin, L. S. Davis, Label consistent K-SVD: Learning a discriminative dictionary for recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11).

[37] I. Kviatkovsky, M. Gabel, E. Rivlin, On the equivalence of the LC-KSVD and the D-KSVD algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2) (2017) 411–416.

[38] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[39] T. Goldstein, S. Osher, The split bregman method for l1-regularized problems, SIAM J. Imaging Sci. 2 (2) (2009) 323–343.

[40] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[41] C. D. Meyer, Matrix Analysis and Applied Linear Algebra, Society for Industrial and Applied Mathematics (SIAM), 2000.

[42] R. Tyrrell Rockafellar, R. J-B Wets, Variational Analysis, Springer, 1998.

[43] H. Attouch, J. Bolte, B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forwardbackward splitting, and regularized Gauss–Seidel methods, Mathematical Programming 137 (1) (2013) 91–129.

[44] P. Ochs, Y. Chen, T. Brox, T. Pock, iPiano: Inertial proximal algorithm for nonconvex optimization, SIAM J. Imag. Sci. 7 (2) (2014) 388–1419.

[45] H. H. Sohrab, Basic Real Analysis, Birkhäuser Basel, 2014.