ℓ_0 norm based dictionary learning by proximal methods with global convergence

Chenglong Bao, Hui Ji, Yuhui Quan and Zuowei Shen
Department of Mathematics, National University of Singapore, Singapore, 119076
{baochenglong,matquan,matjh,matzuows}@nus.edu.sg

Abstract

Sparse coding and dictionary learning have seen their applications in many vision tasks, which usually is formulated as a non-convex optimization problem. Many iterative methods have been proposed to tackle such an optimization problem. However, it remains an open problem to have a method that is not only practically fast but also is globally convergent. In this paper, we proposed a fast proximal method for solving ℓ_0 norm based dictionary learning problems, and we proved that the whole sequence generated by the proposed method converges to a stationary point with sub-linear convergence rate. The benefit of having a fast and convergent dictionary learning method is demonstrated in the applications of image recovery and face recognition.

1. Introduction

In recent years, sparse coding has been widely used in many applications [23], *e.g.*, image recovery, machine learning, and recognition. The goal of sparse coding is to represent given data by the linear combination of few elements taken from a set learned from given training samples. Such a set is called *dictionary* and the elements of the set are called *atoms*. Let $\boldsymbol{D} = \{\boldsymbol{d}_k\}_{k=1}^m \subset \mathbb{R}^n$ denote an over-complete dictionary composed of $m(\geq n)$ atoms. Then, for a signal $\boldsymbol{y} \in \mathbb{R}^n$, its *sparse approximation* over $\boldsymbol{D}$ is about finding a linear expansion $\boldsymbol{Dc} = \sum_{k=1}^m c_k \boldsymbol{d}_k$ using the fewest elements that approximates $\boldsymbol{y}$ with an error bound ϵ. The sparse approximation for an input signal can be formulated as the following optimization problem:

$$\min_{\boldsymbol{c}\in\mathbb{R}^m} \|\boldsymbol{c}\|_0, \quad \text{subject to } \|\boldsymbol{y} - \boldsymbol{Dc}\|_2^2 \leq \epsilon, \tag{1}$$

where $\|\cdot\|_0$ denotes the pseudo-norm that counts the number of non-zeros. The problem (1) is a challenging NP-hard problem and only sub-optimal solutions can be found in polynomial time. Most existing algorithms either use greedy algorithms to iteratively select locally optimal solutions (*e.g.* orthogonal matching pursuit (OMP) [24]), or replace the non-convex ℓ_0 norm by its convex relaxation ℓ_1 norm (*e.g.* basis pursuit [7]).

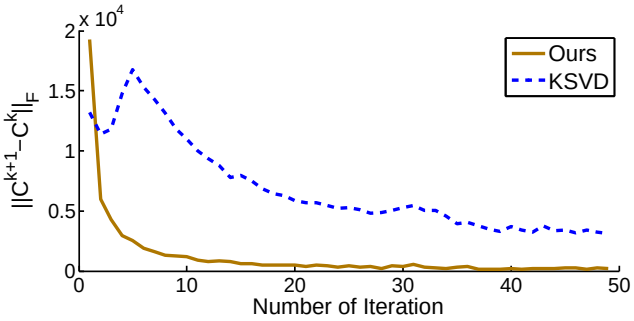


Figure 1. Convergence behavior: the norms of the increments of the coefficient sequence $\boldsymbol{C}^k$ generated by the K-SVD method and the proposed method.

The dictionary for sparse approximation is usually learned from given training samples to maximize the efficiency of sparse approximation in terms of sparsity degree. More concretely, given a training set of p signals $\boldsymbol{Y} := \{\boldsymbol{y}_i\}_{i=1}^p \subset \mathbb{R}^n$, the *dictionary learning* is often formulated as the following minimization problem:

$$\min_{\boldsymbol{D},\{\boldsymbol{c}_k\}_{k=1}^p} \sum_{k=1}^p \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{Dc}_k\|_2^2 + \lambda\|\boldsymbol{c}_k\|_0, \tag{2}$$

subject to $\|\boldsymbol{d}_k\|_2 = 1, k = 1, 2, \ldots, m$, where $\boldsymbol{C} = \{\boldsymbol{c}_k\}_{k=1}^p$ denotes the sparse coefficients of training set Y and $\boldsymbol{D}$ denotes the learned dictionary.

1.1. Motivation

The minimization problem (2) is a non-convex problem whose non-convexity comes from two sources: the sparsity-prompting functional ℓ_0 norm and the bi-linearity between the dictionary $\boldsymbol{D}$ and the codes $\{\boldsymbol{c}_k\}_{k=1}^p$. Most existing approaches (*e.g.* [1, 16, 13, 17]) take an alternating iteration between two modules: sparse approximation for updating $\{\boldsymbol{c}_k\}_{k=1}^p$ and dictionary learning for updating dictionary $\boldsymbol{D}$.

Despite the success of these alternating iterative methods in practice, to best of our knowledge, none of them established the global convergence property, i.e., the whole

sequence generated by the method converges to a stationary point of (2). These schemes can only guarantee that the functional values are decreasing over the iterations, and thus there exists a convergent sub-sequence as the sequence is always bounded. Indeed, the sequence generated by the popular K-SVD method [1] is not convergent as its increments do not decrease to zero. See Fig. 1 for an illustration. The global convergence property is not only of great theoretical importance, but also likely to be more efficient in practical computation as many intermediate results are useless for a method without global convergence property.

1.2. Main contributions

In this paper, we proposed an alternating proximal linearized method for solving (2). The main contribution of the proposed algorithm lies in its theoretical contribution to the open question regarding the convergence property of ℓ_0 norm based dictionary learning methods. In this paper, we showed that the whole sequence generated by the proposed method converges to a stationary point of (2). Moreover, we also showed that the convergence rate of the proposed algorithm is at least sub-linear. To the best of our knowledge, this is the first algorithm with global convergence for solving ℓ_0 norm based dictionary learning problems.

The proposed method can also be used solve other variations of (2) with small modifications, *e.g.* the ones used in discriminative K-SVD based recognition methods [28, 15]. Compared to many existing methods including the K-SVD method, the proposed method also has its advantage on computational efficiency. The experiments showed that the implementation of the proposed algorithm has comparable performance to the K-SVD method in two applications: image de-noising and face recognition, but is noticeably faster.

2. Related work

In this section, we give a brief review on dictionary learning and related applications. Based on the used sparsity prompting functional, the existing dictionary learning methods can be classified into the following three categories.

ℓ_0 norm based methods. The most popular ℓ_0 norm based dictionary learning method is the K-SVD method [1] which used the model (2) for image denoising. Using many image patches from the input image as the training set, the K-SVD method alternatively iterates between sparse approximation and dictionary updating. The sparse approximation is based on the OMP method and the dictionary is estimated via sequential column-wise SVD updates.

The K-SVD method showed good performance in image de-noising and is also used in face/object recognition by adding some additional fidelity term in (2). For example, the so-called discriminative K-SVD method in [28, 15] seeks the sparse code that minimizes both reconstruction error and classification error as follows,

$$\min_{\boldsymbol{D},\boldsymbol{W},\{\boldsymbol{c}_k\}_{k=1}^p} \sum_{k=1}^{p} \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{c}_k\|_2^2 + \sum_{k=1}^{p} \frac{\beta}{2}\|\boldsymbol{l}_k - \boldsymbol{W}\boldsymbol{c}_k\|_2^2, \tag{3}$$

subject to $\|\boldsymbol{c}_j\|_0 \leq \tau, \|\boldsymbol{w}_k\|_2 \leq 1, \|\boldsymbol{d}_k\|_2 \leq 1, j = 1, 2, \ldots, p, k = 1, 2, \ldots, m$, where $\boldsymbol{W} = [\boldsymbol{w}_1, ...\boldsymbol{w}_k]$ denotes the linear classifier learned from the training set and $\boldsymbol{l}_j$ denotes the binary encoded class label of the jth sample. Both dictionary update and sparse approximation is done via calling the K-SVD method. Also using the ℓ_0 norm related optimization model, a fast method is proposed in [6] for learning a tight frame, which has closed form solutions for both sparse approximation and dictionary update.

Convex relaxation methods. As a convex relaxation of ℓ_0 norm, the ℓ_1 norm has been used in many dictionary learning methods to improve the computational feasibility and efficiency of sparse coding; see *e.g.* [16, 13, 17, 26]. All these methods also take an alternating scheme between sparse coding and dictionary updating. In the stage of sparse approximation which requires solving a ℓ_1 norm related minimization problem, various methods have been used in different applications, including the accelerated gradient method [25] or fast iterative shrinkage thresholding algorithm [4] in [13]; the fixed point method [12] in [17]. In the stage of dictionary update, the atoms are either updated one by one or are simultaneously updated. One-by-one atom updating is implemented in [16, 13] as it has closed form solutions. The projection gradient method is used in [17] to update the whole dictionary together. The convergence analysis is provided for the proximal method proposed in [26] for the ℓ_1 norm based dictionary learning.

Non-convex relaxation methods. As shown in [9, 27], the ℓ_1 norm penalty tends to have biased estimation for large coefficients and sometimes results in over-penalization. Thus, several non-convex relaxations of ℓ_0 norm are proposed for better accuracy in sparse coding. For example, the non-convex minimax concave (MC) penalty [27] is used in [21] for sparse dictionary learning. For other non-convex relaxations, *e.g.* smoothly clipped absolute deviation [9] and log penalty [10], the proximal methods have been proposed in [2, 22] to solve the minimization problems with these non-convex regularization terms. The convergence property of these methods is limited to the subsequence convergence.

3. Algorithm and convergence analysis

3.1. Problem formulation

The following definitions and notations are used for discussion. We use bold upper letters for matrices, bold lower letters for column vectors and regular lower letter for elements. For example, $\boldsymbol{y}_j$ denotes the j-th column of the matrix $\boldsymbol{Y}$ and y_i denotes the i-th element of the vector $\boldsymbol{y}$.

For a matrix $\boldsymbol{Y}$, its Frobenius norm is defined as $\|\boldsymbol{Y}\|_F^2 = (\sum_{i,j}|Y_{ij}|^2)^{1/2}$, its ℓ_0 norm $\|\boldsymbol{Y}\|_0$ is defined as the number of nonzero entries in $\boldsymbol{Y}$, and its uniform norm is defined as $\|\boldsymbol{Y}\|_\infty = \max_{i,j}|y_{i,j}|$. Given a matrix $\boldsymbol{Y}$, the *h*ard thresholding operator $T_\lambda(\boldsymbol{Y})$ is defined as $[T_\lambda(\boldsymbol{Y})]_{ij} = \boldsymbol{Y}_{ij}$ if $|\boldsymbol{Y}_{ij}| > \lambda$ and $[T_\lambda(\boldsymbol{Y})]_{ij} = 0$ otherwise.

The original model (2) does not impose any constraint on the code $\{\boldsymbol{c}_k\}$. When a dictionary with high redundancy is adopted, some elements of the sparse coefficient vector could have unusual large values, which in general are not correct. Thus, we slightly modify the model (2) by adding a bound constraint on $\{\boldsymbol{c}_k\}$. Then, the minimization model considered in this paper is defined as follows,

$$\begin{aligned} \min_{\boldsymbol{D},\{\boldsymbol{c}_k\}_{k=1}^p} \quad & \frac{1}{2}\sum_{k=1}^{p}\|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{c}_k\|_2^2 + \lambda\|\boldsymbol{c}_k\|_0 \\ \text{s.t. } & \|\boldsymbol{d}_k\|_2 = 1, 1 \le k \le m; \|\boldsymbol{c}_k\|_\infty < M, 1 \le k \le p, \end{aligned} \tag{4}$$

where M is a pre-defined upper-bound for all elements of $\{\boldsymbol{c}_k\}$. It is noted that the bound constraint on $\{\boldsymbol{c}_k\}_k$ is mainly for improving the stability of the model (2), which can be set to a sufficiently large value to avoid any negative impact on the accuracy of the coefficients. For the simplicity of discussion, let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p]$ denote the training sample matrix and let $C = [\boldsymbol{c}_1, \ldots \boldsymbol{c}_p]$ denote the coefficient matrix. Let $\mathcal{X} = \{\boldsymbol{D} \in \mathbb{R}^{n\times m} : \|\boldsymbol{d}_k\|_2 = 1, 1 \le k \le m\}$ denote the feasible set for the dictionary $\boldsymbol{D}$, and let $\mathcal{C} = \{\boldsymbol{C} \in \mathbb{R}^{m\times p} : \|\boldsymbol{c}_k\|_\infty \le M, 1 \le k \le p\}$ denote the feasible set for the coefficient matrix $\boldsymbol{C}$. Then the model (4) can be expressed in the following compact form:

$$\min_{\substack{\boldsymbol{D}\in\mathbb{R}^{n\times m}\\ \boldsymbol{C}\in\mathbb{R}^{m\times p}}} \quad \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_0, \tag{5}$$

subject to $\boldsymbol{D} \in \mathcal{X}, \boldsymbol{C} \in \mathcal{C}$. In the next, we will present an alternating proximal method for solving (5), as well as provide the convergence analysis.

3.2. Alternating proximal method

The proposed algorithm is based on the proximal method [14] for solving the following non-convex problem:

$$\min_{\boldsymbol{x},\boldsymbol{y}} H(\boldsymbol{x},\boldsymbol{y}) = F(\boldsymbol{x}) + Q(\boldsymbol{x},\boldsymbol{y}) + G(\boldsymbol{y}), \tag{6}$$

where $F(\boldsymbol{x}), G(\boldsymbol{y})$ are proper lower semi-continuous functions, and $Q(\boldsymbol{x},\boldsymbol{y})$ is a smooth function with Lipschitz gradient on any bounded set. The proximal method proposed in [14] updates the estimate of $(\boldsymbol{x},\boldsymbol{y})$ via solving the following proximal problems:

$$\begin{aligned} \boldsymbol{x}^{k+1} \in \arg\min_{\boldsymbol{x}} & F(\boldsymbol{x}) + \langle \boldsymbol{x} - \boldsymbol{x}^k, \nabla_{\boldsymbol{x}} Q(\boldsymbol{x}^k, \boldsymbol{y}^k)\rangle \\ & + \frac{t_k^1}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|_2^2; \\ \boldsymbol{y}^{k+1} \in \arg\min_{\boldsymbol{y}} & G(\boldsymbol{y}) + \langle \boldsymbol{y} - \boldsymbol{y}^k, \nabla_{\boldsymbol{y}} Q(\boldsymbol{x}^{k+1}, \boldsymbol{y}^k)\rangle \\ & + \frac{t_k^2}{2}\|\boldsymbol{y} - \boldsymbol{y}^k\|_2^2, \end{aligned} \tag{7}$$

where t_k^1 and t_k^2 are two appropriately chosen step sizes. Using the so-called *proximal operator* [19] defined as

$$\text{Prox}_t^F(\boldsymbol{x}) := \arg\min_{\boldsymbol{u}} F(\boldsymbol{u}) + \frac{t}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_2^2, \tag{8}$$

the minimizations (7) are equivalent to the following proximal problem:

$$\begin{aligned} \boldsymbol{x}^{k+1} &\in \text{Prox}_{t_k^1}^F(\boldsymbol{x}^k - \frac{1}{t_k^1}\nabla Q(\boldsymbol{x}^k, \boldsymbol{y}_k)), \\ \boldsymbol{y}^{k+1} &\in \text{Prox}_{t_k^2}^G(\boldsymbol{y}^k - \frac{1}{t_k^2}\nabla Q(\boldsymbol{x}^{k+1}, \boldsymbol{y}_k)). \end{aligned} \tag{9}$$

Remark Without the convexity assumption, it is shown in [14] that for any proper and lower semicontinuous function bounded below, the proximal map (8) is nonempty and compact for any $t \in (0, +\infty)$.

The minimization problem (5) can be expressed in the form of (6) by setting

$$\begin{cases} F(\boldsymbol{C}) = \|\boldsymbol{C}\|_0 + I_{\mathcal{C}}(\boldsymbol{C}); \\ Q(\boldsymbol{C},\boldsymbol{D}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2; \\ G(\boldsymbol{D}) = I_{\mathcal{X}}(\boldsymbol{D}), \end{cases} \tag{10}$$

where $I_{\mathcal{X}}(\boldsymbol{D})$ denotes the indicator function of $\boldsymbol{D}$ that satisfies $I_{\mathcal{X}}(\boldsymbol{D}) = 0$ if $\boldsymbol{D} \in \mathcal{X}$ and $+\infty$ otherwise. Then using proximal operators, we propose the following alternating iterative scheme for solving (5): let $\boldsymbol{D}^{(0)}$ be the initial dictionary, then for $\ell = 0, 1, \cdots,$

1. **sparse approximation:** given the dictionary $\boldsymbol{D}^{(\ell)}$, find the sparse code $\boldsymbol{C}^{(\ell)}$ that satisfies

$$\boldsymbol{C}^{(\ell)} \in \text{Prox}_{\lambda_\ell/\lambda}^F(\boldsymbol{C}^{(\ell-1)} - \frac{1}{\lambda_\ell}\nabla_{\boldsymbol{C}} Q(\boldsymbol{C}^{(\ell-1)}, \boldsymbol{D}^{(\ell)})), \tag{11}$$

   where λ_ℓ is an estimated step size (more on this later).

2. **dictionary update:** given the sparse code $\boldsymbol{C}^{(\ell)}$, update the dictionary $\boldsymbol{D}^{(\ell+1)} = \{\boldsymbol{d}_k^{(\ell+1)}\}_{k=1}^m$ atom by atom:

$$\boldsymbol{d}_k^{(l+1)} \in \text{Prox}_{\mu_\ell^k}^{G(\hat{\boldsymbol{D}}_k^{(\ell)})}(\boldsymbol{d}_k^{(\ell)} - \frac{1}{\mu_l^k}\nabla_{\boldsymbol{d}_k} Q(\boldsymbol{C}^{(\ell+1)}), \widetilde{\boldsymbol{D}}^{(\ell)})), \tag{12}$$

where

$$\begin{cases} \hat{\boldsymbol{D}}_k^{(\ell)} = [\boldsymbol{d}_1^{(\ell+1)}, \cdots, \boldsymbol{d}_{k-1}^{(\ell+1)}, \boldsymbol{d}_k, \boldsymbol{d}_{k+1}^{(\ell)}, \cdots, \boldsymbol{d}_n^{(\ell)}]; \\ \widetilde{\boldsymbol{D}}_k^{(\ell)} = [\boldsymbol{d}_1^{(\ell+1)}, \cdots, \boldsymbol{d}_{k-1}^{(\ell+1)}, \boldsymbol{d}_k^{(\ell)}, \boldsymbol{d}_{k+1}^{(\ell)}, \cdots, \boldsymbol{d}_n^{(\ell)}], \end{cases}$$

and μ_k^ℓ is a step size need to be estimated.

Each iteration above requires solving two optimization problems (11) and (12). In the next, we show that both have closed form solutions. Define

$$\begin{cases} \boldsymbol{T}_{\boldsymbol{C}}^{(\ell)} = \boldsymbol{C}^{(\ell-1)} - \frac{1}{\lambda_\ell} \nabla_{\boldsymbol{C}} Q(\boldsymbol{C}^{(\ell-1)}, \boldsymbol{D}^{(\ell)}); \\ \boldsymbol{S}_k^{(\ell)} = \boldsymbol{d}_k^{(\ell)} - \frac{1}{\mu_k^\ell} \nabla_{\boldsymbol{d}_k} Q(\boldsymbol{C}^{(\ell)}, \widetilde{\boldsymbol{D}}_k^{(\ell)}). \end{cases}$$

Then by a direct calculation, two optimization problems (11) and (12) are equivalent to

$$\begin{cases} \boldsymbol{C}^{(\ell)} \in \arg\min_{\boldsymbol{C} \in \mathcal{C}} \frac{\lambda_\ell}{2\lambda} \|\boldsymbol{C} - \boldsymbol{T}_{\boldsymbol{C}}^{(\ell)}\|_F^2 + \|\boldsymbol{C}\|_0, \\ \boldsymbol{d}_k^{(\ell)} \in \arg\min_{\|\boldsymbol{d}_k\|_2 = 1} \frac{1}{2} \|\boldsymbol{d}_k - \boldsymbol{S}_k^{(\ell)}\|, 1 \le k \le m. \end{cases} \tag{13}$$

Proposition 3.1 *Suppose that M is chosen such that $M > \sqrt{2\lambda/\lambda_\ell}$, two minimization problems in (13) have the closed form solutions given by*

$$\begin{cases} \boldsymbol{C}^{(\ell)} = \min\{T_{\sqrt{2\lambda/\lambda_\ell}}(\boldsymbol{T}_{\boldsymbol{C}}^{(\ell)}), M\}; \\ \boldsymbol{d}_k^{(\ell)} = \boldsymbol{S}_k^{(\ell)} / \|\boldsymbol{S}_k^{(\ell)}\|_2, 1 \le k \le m. \end{cases} \tag{14}$$

Proof The proof of the solution to the second problem in (13) is trivial. The first is easy to obtain as it can be decomposed into the summation of independent minimization problems with respect to each variable.

Setting of step sizes. There are two step sizes, λ_ℓ in (11) and μ_ℓ^k in (12), need to be set during the iteration. The step size λ_ℓ can be chosen as $\lambda_\ell = \max\{\rho L(\boldsymbol{D}^{(\ell)}), \underline{\ell}\}$ where $\underline{\ell} > 0$ is a constant, $\rho > 1$ and $L(\boldsymbol{D}^{(\ell)})$ satisfies

$$\|\nabla_{\boldsymbol{C}}(Q(\boldsymbol{C}_1, \boldsymbol{D}^{(\ell)})) - \nabla_{\boldsymbol{C}} Q(\boldsymbol{C}_2, \boldsymbol{D}^{(\ell)})\| \le L(\boldsymbol{D}^{(\ell)}) \|\boldsymbol{C}_1 - \boldsymbol{C}_2\|.$$

The step size μ_l^k can be chosen as $\mu_l^k = \max\{\rho L(\boldsymbol{z}_k^{(\ell)}), \underline{\ell}\}$ where $\boldsymbol{z}^{(\ell)} = (\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)}) - \boldsymbol{d}_k^{(\ell)}$, $\underline{\ell} > 0$, $\rho > 1$ and $L(\boldsymbol{z}_k^{(\ell)})$ satisfies

$$\|\nabla_{\boldsymbol{d}_k}(Q(\boldsymbol{z}^{(\ell)}, \boldsymbol{d}_k^1) - \nabla_{\boldsymbol{C}} Q(\boldsymbol{z}^{(\ell)}, \boldsymbol{d}_k^2)\| \le L(\boldsymbol{z}_k^{(\ell)}) \|\boldsymbol{d}_k^1 - \boldsymbol{d}_k^2\|,$$

for any pair $\boldsymbol{d}_k^1, \boldsymbol{d}_k^2$. Consequently, we can choose $L(\boldsymbol{D}^\ell) = \|\boldsymbol{D}^{(\ell)^\top} \boldsymbol{D}^{(\ell)}\|_F$ and $L(\boldsymbol{z}_k^{(\ell)}) = [\boldsymbol{C}^{(\ell)} \boldsymbol{C}^{(\ell)\top}]_{k,k}, \forall k = 1, 2, \cdots, m$. It can be seen that the sequence $L(\boldsymbol{D}^{(\ell)})$ is a bounded sequence since each column in $\boldsymbol{D}$ is of unit norm. Moreover, the sequence $L(\boldsymbol{z}_k^{(\ell)})$ is also a bounded sequence since both $\boldsymbol{C}$ and $\boldsymbol{D}$ are bounded. See Alg.1 for the outline of the proposed dictionary learning method that solves (5).

Iteration complexity. The main computational cost of our algorithm 1 lies in the matrix product in the sparse coding stage. So, the algorithm 1 has $O(mnp)$ iteration complexity which is less than $O(mnp + K^2 mp)$, the iteration complexity of the accelerated version of the K-SVD method [20], where K is the predefined sparsity level.

Algorithm 1 Proximal method for dictionary learning

1: **INPUT**: Training signals $\boldsymbol{Y}$
2: **OUTPUT**: Learned Dictionary $\boldsymbol{D}$
3: **Main Procedure:**
1. Initialization: set dictionary $\boldsymbol{D}^{(0)}$, $\rho > 1, \underline{\ell} > 0$.
2. For $\ell = 0, 1, \cdots$
(a) Sparse approximation:

$$\begin{aligned} &\lambda_\ell = \max\{\rho L(\boldsymbol{D}^{(\ell)}), \underline{\ell}\}; \\ &\boldsymbol{T}_{\boldsymbol{C}}^{(\ell)} = \boldsymbol{C}^{(\ell-1)} - \frac{1}{\lambda_\ell} \nabla_{\boldsymbol{C}} Q(\boldsymbol{C}^{(\ell-1)}, \boldsymbol{D}^{(\ell)}); \\ &\boldsymbol{C}^{(\ell)} = \min\{T_{\sqrt{2\lambda/\lambda_\ell}}(\boldsymbol{T}_{\boldsymbol{C}}^{(\ell)}), M\}. \end{aligned} \tag{15}$$

(b) for $k = 1, \cdots, m$,

$$\boldsymbol{V}^{(\ell)} = \boldsymbol{C}^{(\ell)} \boldsymbol{C}^{(\ell)\top}, \quad L(\boldsymbol{z}_k^{(\ell)}) = \boldsymbol{V}_{k,k}^{(\ell)}.$$

(c) Dictionary update: for $k = 1, \cdots, p$,

$$\begin{aligned} &\mu_l^k = \max\{\rho L(\boldsymbol{z}_k^{(\ell)}), \underline{\ell}\}; \\ &\boldsymbol{S}_k^{(\ell)} = \boldsymbol{d}_k^{(\ell)} - \frac{1}{\mu_l^k} \nabla_{\boldsymbol{d}_k} Q(\boldsymbol{C}^{(\ell)}, \widetilde{\boldsymbol{D}}_k^{(\ell)}); \\ &\boldsymbol{d}_k^{(\ell+1)} = \boldsymbol{S}_k^{(\ell)} / \|\boldsymbol{S}_k^{(\ell)}\|_2. \end{aligned} \tag{16}$$

(d) $L(\boldsymbol{D}^{(\ell+1)}) = \|D^{(\ell+1)\top} D^{(\ell+1)}\|_F$.

Remark Algorithm 1 can be further accelerated by updating its associated coefficients right after one dictionary item is updated. The coefficient update can be done using least squares regression on the same support of the previous one.

4. Global convergence of Algorithm 1

Before proving the global convergence of Alg. 1, we first introduce the definition of the critical points of a non-convex function given in [14].

Definition Given the non-convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper and lower semi-continuous function and $\mathrm{dom} f = \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty\}$.

- For $\boldsymbol{x} \in \mathrm{dom} f$, its *Frechet subdifferential* of f is defined as

$$\hat{\partial} f(\boldsymbol{x}) = \{\boldsymbol{u} : \liminf_{\boldsymbol{y} \to \boldsymbol{x}, \boldsymbol{y} \neq \boldsymbol{x}} \frac{f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{u}, \boldsymbol{y} - \boldsymbol{x} \rangle}{\|\boldsymbol{y} - \boldsymbol{x}\|} \ge 0\}$$

and $\hat{\partial} f(\boldsymbol{x}) = \emptyset$ if $\boldsymbol{x} \notin \mathrm{dom} f$.

- The *Limiting Sub-differential* of f at $\boldsymbol{x}$ is defined as

$$\partial f(\boldsymbol{x}) = \{\boldsymbol{u} \in \mathbb{R}^n : \exists \boldsymbol{x}^k \to \boldsymbol{x}, f(\boldsymbol{x}^k) \to f(\boldsymbol{x}) \text{ and } \boldsymbol{u}^k \in \hat{\partial} f(\boldsymbol{x}^k) \to \boldsymbol{u}\}$$

- The point $\boldsymbol{x}$ is a *critical point* of f if $0 \in \partial f(\boldsymbol{x})$.

Remark More on critical points of (5).

- If $\boldsymbol{x}$ is a local minimizer of f then $0 \in \partial f(\boldsymbol{x})$.
- If $(\boldsymbol{C}, \boldsymbol{D})$ is the critical point of (5), then we have

$$(\boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{C})[i,j] = (\boldsymbol{D}^\top \boldsymbol{Y})[i,j] \text{ if } \boldsymbol{C}[i,j] \neq 0.$$

Theorem 4.1 *[Global convergence] The sequence generated by the algorithm 1, $\{(\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)})\}$, is a Cauchy sequence and converges to a critical point of (5).*

Proof See Appendix A.

Remark Different from the subsequence convergence property, the global convergence property is defined as: $(\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)}) \to (\bar{\boldsymbol{C}}, \bar{\boldsymbol{D}})$, as $\ell \to +\infty$.

Next, we show that Algorithm 1 has at least of sublinear convergent rate.

Theorem 4.2 *[Sub-linear convergence rate] The sequence generated by the Alg. 1, $\{(\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)})\}$, converges to a critical point $(\bar{\boldsymbol{C}}, \bar{\boldsymbol{D}})$ of (5) at least in the sublinear convergence rate, i.e. there exist some $\omega > 0$, such that*

$$\|(\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)}) - (\bar{\boldsymbol{C}}, \bar{\boldsymbol{D}})\| \leq \omega \ell^{\frac{1-\theta}{2\theta-1}} \tag{17}$$

where $\theta \in (\frac{1}{2}, 1)$.

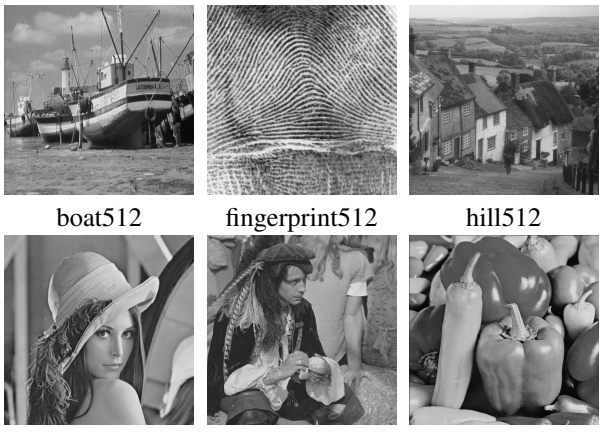
Proof See Appendix B.

5. Experiments

In this section, the practical performance and computational efficiency of the proposed approach is evaluated on two applications: image de-noising and face recognition. The experiments on these two applications showed that, using the same minimization model, the performance of our approach is comparable to the K-SVD based method, but is more computationally efficient with less running time.

5.1. Image Denoising

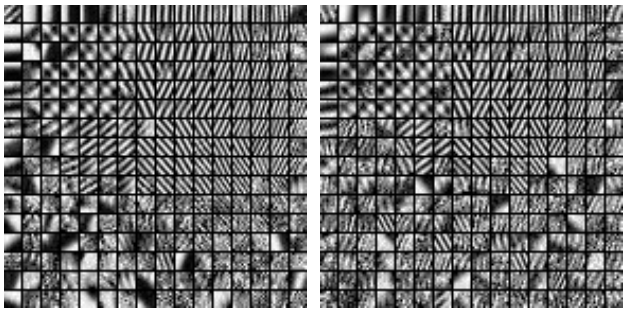
Alg. 1 for image denoising is evaluated on tested images shown in Fig. 2 with different noise levels σ. Through all the experiments, we set $\lambda = (0.03\sigma)^2/2$ as the thresholding value for dictionary learning process. Same as the K-SVD method [8], the dictionary is equipped with $m = 4n$ atoms and initialized with overcomplete DCT dictionary.



boat512 fingerprint512 hill512

Lena512 man512 peppers512

Figure 2. Test images.



(e) K-SVD (f) Alg.1

Figure 3. The dictionaries learned from the image "Lena512" with noise level $\sigma = 30$ using the K-SVD method and Alg.1. The atom size is 8×8.

The maximum iteration of Alg. 1 is set 30. After the dictionary is learned via training samples, the image is denoised using the coefficients from the OMP method under the learned dictionary in one pass. The results are compared to the DCT-based thresholding method and the K-SVD denoising method [8] with patch size 8×8. See Table 1 for the list of PSNR values of the results and Fig. 4 for a visual illustration of the denoised images. Fig. 3 shows the dictionaries learned from noisy image by both the K-SVD method and the proposed method. It can be seen that the performance of our approach is comparable to the K-SVD method.

The computational efficiency of the proposed one is compared to the accelerated version of the K-SVD method (*i.e.* the approximated K-SVD Algorithm [20] with the implementation from the original authors[1]). All two methods run on the same environment: MATLAB R2011b (64bit) Linux version on a PC workstation with an INTEL CPU (2.4GHZ) and 48G memory. The average running time of each iteration is: 1.81 seconds (K-SVD) vs. 0.11 seconds (ours). Fig. 5 shows the comparison of the overall running

[1] http://www.cs.technion.ac.il/ ronrubin/software.html

noisy image ($\sigma = 25$) restored image

Figure 4. Visual illustration of noisy images and denoised results

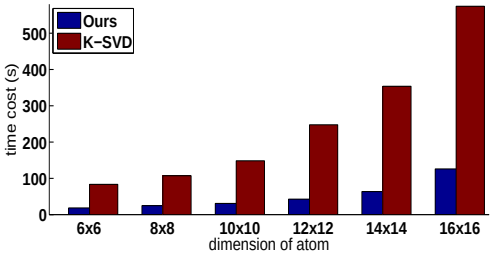


Figure 5. Overall running time of our method and the K-SVD denoising method with comparable PSNR values.

time of the accelerated implementation of Alg.1 and the K-SVD method to denoise image "Lena512" with noise level $\sigma = 25$. Clearly, Alg.1 is noticeably faster than the approximate K-SVD method when learning the dictionary of the same size. More importantly, Alg.1 is more scalable for high-dimensional data.

5.2. Face Recognition

Alg. 1 can also be applied to recognition tasks using the model (3) by simply replacing the K-SVD module by the proposed one. The performance is evaluated on two face datasets: Extended YaleB dataset [11] and AR face dataset [18]. The one used our approach is compared to three K-SVD based methods: LC-KSVD [15], D-KSVD [28] and K-SVD [1]. The experimental setting is set the same as [28, 15]:

Extended YaleB Dataset: The extended YaleB dataset [11] contains 2,414 images of 38 human frontal faces under about 64 illumination conditions and expressions. There are about 64 images for each person. The original images were cropped to 192×168 pixels. Following [28], we project each face image into a 504-dimensional feature vector using a random matrix of zero-mean normal distribution. The database is randomly split into two halves. One half which contains 32 images for each person was used for training the dictionary. The other half was used for testing.

AR Face Dataset: The AR face dataset [18] consists of over 4000 frontal images from 126 individuals. For each individual, 26 pictures were taken in two separate sessions. The main characteristic of the AR database is that it includes frontal views of faces with different facial expressions, lighting conditions and occlusion conditions. Following the standard evaluation procedure from [28, 15], we use a subset of the database consisting of 2,600 images from 50 male subjects and 50 female subjects. For each person, twenty images are randomly picked up for training and the remaining images are for testing. Each face image is cropped to 165×120 and then projected onto a 540-dimensional feature vector.

We set the thresholding parameter λ to be $10^{-4}/2$ and initialize the dictionary with identity matrix. Besides the classification accuracies, we also evaluate the training time of all compared approaches under the same environment. The results of all the tested methods are listed in Table 3 and Table 2. It can be seen that our approach performs consistently with the state-of-the-art methods while has noticeable advantages on computational efficiency.

6. Summary

In this paper, we proposed an alternating proximal method iteration scheme for solving ℓ_0 norm based dictionary learning problems in sparse coding. The proposed one not only answered the open question regarding the existence of a convergent method for solving ℓ_0 norm based dictionary learning problems, but also showed the computational efficiency on two practical applications. In future, we will investigate the applications of the proposed framework for solving other non-convex minimization problems in computer vision.

Acknowledgment

The authors would like to thank the area chair and the reviewers for their helpful comments. The work of the authors was partially supported by Singapore MOE Research Grant R-146-000-165-112 and R-146-000-178-112. Yuhui Quan would like to help provided by Yuping Sun on the experiments.

References

[1] M. Aharon and M. E. Bruckstein. K-SVD: An algorithm for designing of overcomplete dictionaries for sparse represen-

Image	Boat512					Fingerprint512					Hill512				
σ	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
DCT; 8×8	36.79	33.49	31.34	29.96	28.90	36.34	32.25	29.68	28.29	26.85	36.54	32.93	31.11	30.02	29.00
K-SVD; 8×8	37.17	33.64	31.73	30.36	29.28	36.59	32.39	30.06	28.47	27.26	36.99	33.34	31.43	30.17	29.19
Ours; 8×8	37.02	33.57	31.62	30.20	29.16	36.59	32.35	29.97	28.28	27.03	36.94	33.31	31.29	30.02	29.06
Image	Lena512					Man512					Peppers512				
σ	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
DCT; 8×8	38.29	35.25	33.39	32.03	30.96	37.16	33.12	31.01	29.65	28.67	37.06	34.48	33.02	31.89	30.95
K-SVD; 8×8	38.59	35.47	33.70	32.38	31.32	37.61	33.62	31.45	30.13	29.11	37.77	34.72	32.37	32.26	31.39
Ours; 8×8	38.49	35.41	33.57	32.25	31.19	37.46	33.47	31.43	30.02	29.00	37.68	34.64	33.22	32.14	31.18

Table 1. PSNR values of the denoised results

Table 2. Training time (seconds) on two face datasets.

Dataset	K-SVD	D-KSVD	LC-KSVD	Ours
Extended YaleB	44.46	63.47	184.64	10.52
AR Face	55.03	70.43	256.12	22.75

Table 3. Classification accuracies (%) on two face datasets.

Dataset	K-SVD	D-KSVD	LC-KSVD	Ours
Extended YaleB	93.10	94.10	95.00	95.66
AR Face	86.50	88.80	93.70	94.41

tation. *IEEE Trans. Signal Process.*, 2006. 1, 2, 6

[2] A.Rakotomamonjy. Direct optimization of the dictionary learning. *IEEE Trans. Signal Process.*, 2013. 2

[3] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 2009. 8

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci.*, 2009. 2

[5] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM J Optimiz*, 2007. 8

[6] J. Cai, H. Ji, Z. Shen, and G. Ye. Data-driven tight frame construction and image denoising. *Appl. Comp. Harm. Anal.*, In Press, 2014. 2

[7] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 1999. 1

[8] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process*, 2006. 5

[9] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Assoc.*, 2001. 2

[10] J. H. Friedman. Fast sparse regression and classification. *International Journal of Forecasting*, 2012. 2

[11] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE PAMI*, 2001. 6

[12] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. *CAAM Report*, 2007. 2

[13] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010. 1, 2

[14] B. Jerome, S. Shoham, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 2013. 3, 4, 7, 8

[15] Z. Jiang, Z. Lin, and L. Davis. Learning a dicscriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011. 2, 6

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 2010. 1, 2

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2009. 1, 2

[18] A. Martínez and R. Benavente. The AR face database. Technical report, Computer Vision Center, 1998. 6

[19] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis: grundlehren der mathematischen wissenschaften*, volume 317. Springer, 1998. 3

[20] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008. 4, 5

[21] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang. A non-convex relaxation approach to sparse dictionary learning. In *In CVPR*, 2011. 2

[22] S. Sra. Scalable nonconvex inexact proximal splitting. In *NIPS*, 2012. 2

[23] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Process. Mag.*, 2011. 1

[24] A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 2004. 1

[25] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM J. Optimiz.*, 2008. 2

[26] Y. Xu and W. Yin. A fast patch-dictionary method for the whole image recovery. *UCLA CAM report*, 2013. 2

[27] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 2010. 2

[28] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, 2010. 2, 6

Appendix A. The proof of Theorem 4.1 is built upon Theorem 1 from [14].

Theorem 6.1 *[14] The sequence $\boldsymbol{Z}^\ell = (\boldsymbol{x}^{(\ell)}, \boldsymbol{y}^{(\ell)})$ generated by the iteration (7) converges to the critical point of (6), if the following conditions hold:*

1. *$H(\boldsymbol{x}, \boldsymbol{y})$ is a KL function;*
2. *$\boldsymbol{Z}^{(\ell)}, \ell = 1, 2, \cdots$ is a bounded sequence and there exists some positive constant $\underline{\ell}, \bar{\ell}$ such that $t_\ell^1, t_\ell^2 \in (\underline{\ell}, \bar{\ell}), \ell = 1, 2, \cdots$;*
3. *$\nabla Q(\boldsymbol{x}, \boldsymbol{y})$ has Lipschitz constant on any bounded set.*

The first condition requires that the objective function satisfies the so-called *Kurdyka-Lojasiewicz (KL)* properties in its effective domain; see Definition 3 in [14] for more details on KL properties. It is shown in Remark 5 and Theorem 11 in [5] that any so-called *semi-algebraic* function satisfy the Kurdyka-Lojasiewicz property. In the next, we first give the definition of the semi-algebraic sets and functions, followed by the proof that the objective function (6) defined via (10) is a semi-algebraic function.

Definition [14] A subset S of $\mathbb{R}^n$ is called the semi-algebraic set if there exists a finite number of real polynomial functions g_{ij}, h_{ij} such that

$$S = \bigcup_j \bigcap_i \{\boldsymbol{u} \in \mathbb{R}^n : g_{ij}(\boldsymbol{u}) = 0, h_{ij}(\boldsymbol{u}) < 0\}.$$

A function $f(\boldsymbol{u})$ is called the semi-algebraic function if its graph $\{(\boldsymbol{u}, t) \in \mathbb{R}^n \times \mathbb{R}, t = f(\boldsymbol{u})\}$ is a semi-algebraic set.

The next lemma establishes that the objective function (6) defined via (10) is a semi-algebraic function.

Lemma 6.2 *Each term in the function (6) defined via (10) is a semi-algebraic function, and thus the function (6) defined via (10) is a semi-algebraic function.*

Proof For $Q(\boldsymbol{C}, \boldsymbol{D}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2$ is a real polynomial function, $Q(\boldsymbol{C}, \boldsymbol{D})$ is a semi-algebraic function [14].

It is easy to notice that the set $\mathcal{X} = \{\boldsymbol{Y} \in \mathbb{R}^{n\times m} : \|\boldsymbol{y}_k\|_2 = 1, 1 \le k \le m\} = \bigcap_{k=1}^m \{\boldsymbol{Y} : \sum_{j=1}^n \boldsymbol{y}_{kj}^2 = 1\}$ is a semi-algebraic set. And the set $\mathcal{C} = \{\boldsymbol{C} \in \mathbb{R}^{m\times p} | \|\boldsymbol{c}_k\|_\infty \le M\} = \bigcup_{j=1}^M \bigcup_{k=1}^p \{\boldsymbol{C} : \|\boldsymbol{c}_k\|_\infty = j\}$ is a semi-algebraic set. Therefore, the indicator functions $I_{\mathcal{C}}(\boldsymbol{C})$ and $I_{\mathcal{X}}(\boldsymbol{D})$ are semi-algebraic functions from the fact that the indicator function for semi-algebraic sets are semi-algebraic functions [3].

For the function $F(\boldsymbol{C}) = \|\boldsymbol{C}\|_0$. The graph of F is $S = \bigcup_{k=0}^{mp} L_k \triangleq \{(\boldsymbol{C}, k) : \|\boldsymbol{C}\|_0 = k\}$. For each $k = 0, \cdots, mp$, let $\mathcal{S}_k = \{J : J \subseteq \{1, \cdots, mp\}, |J| = k\}$, then $L_k = \bigcup_{J\in\mathcal{S}_k} \{(\boldsymbol{C}, k) : \boldsymbol{C}_{J^c} = 0\}$. It is easy to know the set $\{(\boldsymbol{C}, k) : \boldsymbol{C}_{J^c} = 0\}$ is a semi-algebraic set in $\mathbb{R}^{m\times p}\times\mathbb{R}$. Thus, $F(\boldsymbol{C}) = \|\boldsymbol{C}\|_0$ is a semi-algebraic function since the finite union of the semi-algebraic set is still semi-algebraic.

For the second condition in theorem 6.1, $\boldsymbol{C}^{(\ell)} \in \mathcal{C}$ and $\boldsymbol{D}^{(\ell)} \in \mathcal{X}$ for any $\ell = 1, 2, \cdots$, which implies $\boldsymbol{Z}^\ell = (\boldsymbol{C}^{(\ell)}, \boldsymbol{D}^{(\ell)})$ is a bounded sequence. In addition, for $\ell = 1, 2\cdots$, the step size $\lambda_\ell = \max(\rho L(\boldsymbol{D}^{(\ell)}), \underline{l})$ is bounded above since $L(\boldsymbol{D}^{(\ell)}) = \|\boldsymbol{D}^{(\ell)\top}\boldsymbol{D}^{(\ell)}\|_F$ and $\boldsymbol{D} \in \mathcal{X}$. The same holds for the step size $\{\mu_k^\ell\}_{k=1}^m$ since $\mu_k^\ell = \max(\rho L(\boldsymbol{z}_k^{(\ell)}), \underline{l})$ where $L(\boldsymbol{z}_k^{(\ell)}) = [\boldsymbol{C}^{(\ell)}\boldsymbol{C}^{(\ell)\top}]_{k,k}$ is bounded above. Consequently, there exists $\underline{l}, \bar{l} > 0$ such that $\lambda_\ell, \mu_k^\ell \in (\underline{l}, \bar{l})$ for any k, ℓ.

For the last condition in theorem 6.1, notice that the function $Q(\boldsymbol{C}, \boldsymbol{D}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2$ is a smooth function. More specifically, $\nabla Q(\boldsymbol{C}, \boldsymbol{D}) = (\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{C} - \boldsymbol{Y}), (\boldsymbol{D}\boldsymbol{C} - \boldsymbol{Y})\boldsymbol{C}^\top)$ has Lipschitz constant on any bounded set. In other words, for any bounded set $\mathcal{M}$, there exists a constant $M > 0$, such that for any $\{(\boldsymbol{C}_1, \boldsymbol{D}_1), (\boldsymbol{C}_2, \boldsymbol{D}_2)\} \subseteq \mathcal{M}$,

$$\|\nabla Q(\boldsymbol{C}_1, \boldsymbol{D}_1) - \nabla Q(\boldsymbol{C}_2, \boldsymbol{D}_2)\| \le M\|(\boldsymbol{C}_1, \boldsymbol{D}_1) - (\boldsymbol{C}_2, \boldsymbol{D}_2)\|.$$

Appendix B. The proof of Theorem 4.2 is a direct application of the following theorem established in [3].

Proposition 6.3 ([3]) *For a given semi-algebraic function $f(\boldsymbol{u})$, for all $\boldsymbol{u} \in \mathrm{dom} f$, there exists $\theta \in [0, 1)$, $\eta \in (0, +\infty]$ a neighborhood U of $\boldsymbol{u}$ and a concave and continuous function $\phi(s) = cs^{1-\theta}, s \in [0, \eta)$ such that for all $\bar{\boldsymbol{u}} \in U$ and satisfies $f(\bar{\boldsymbol{u}}) \in (f(\boldsymbol{u}), f(\boldsymbol{u}) + \eta)$, the following inequality holds*

$$\phi'(f(\bar{\boldsymbol{u}}) - f(\boldsymbol{u}))\mathrm{dist}(0, \partial f(\bar{\boldsymbol{u}})) \ge 1 \tag{18}$$

where $\mathrm{dist}(0, \partial f(\bar{\boldsymbol{u}}) = \max\{\|\boldsymbol{u}^*\| : \boldsymbol{u}^* \in \partial f(\bar{\boldsymbol{u}})\}$.

Theorem 6.4 ([3]) *If the objective function is semi-algebraic, $\boldsymbol{Z}^\ell = (\boldsymbol{x}^{(\ell)}, \boldsymbol{y}^{(\ell)})$ generated by the iteration (7), and $\bar{\boldsymbol{Z}} = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is its limit point. Then*

- *If $\theta = 0$, $\boldsymbol{Z}^\ell$ converges to $\bar{\boldsymbol{Z}}$ in finite steps.*
- *If $\theta \in (0, 1/2]$, then $\exists \omega > 0$ and $\tau \in [0, 1)$, such that $\|\boldsymbol{Z}^\ell - \bar{\boldsymbol{Z}}\| \le \omega\tau^\ell$*
- *If $\theta \in (1/2, 1)$, then $\exists \omega > 0$ such that $\|\boldsymbol{Z}^\ell - \bar{\boldsymbol{Z}}\| \le \omega\ell^{-\frac{1-\theta}{2\theta-1}}$.*

where θ corresponding to the desingularizing function $\phi(s) = cs^{1-\theta}$ defined in proposition 6.3.

In the proposed Alg.1, notice that $\frac{\tau^\ell}{\ell^{-\frac{1-\theta}{2\theta-1}}} \to 0$ as $\ell \to +\infty$, where $\tau \in [0, 1)$ and $\theta \in (1/2, 1)$. Thus, the sequence $\boldsymbol{Z}^\ell$ converges to $\bar{\boldsymbol{Z}}$ at least in sub-linear rate.