# A Convergent Incoherent Dictionary Learning Algorithm for Sparse Coding

Chenglong Bao, Yuhui Quan, Hui Ji

Department of Mathematics
National University of Singapore

**Abstract.** Recently, sparse coding has been widely used in many applications ranging from image recovery to pattern recognition. The low mutual coherence of a dictionary is an important property that ensures the optimality of the sparse code generated from this dictionary. Indeed, most existing dictionary learning methods for sparse coding either implicitly or explicitly tried to learn an incoherent dictionary, which requires solving a very challenging non-convex optimization problem. In this paper, we proposed a hybrid alternating proximal algorithm for incoherent dictionary learning, and established its global convergence property. Such a convergent incoherent dictionary learning method is not only of theoretical interest, but also might benefit many sparse coding based applications.

**Keywords:** mutual coherence, dictionary learning, sparse coding

## 1 Introduction

Recently, sparse coding has been one important tool in many applications ([24]) including image recovery, machine learning, recognition and etc. Given a set of input patterns, most existing sparse coding models aim at finding a small number of *atoms* (representative patterns) whose linear combinations approximate those input patterns well. More specifically, given a set of vectors $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_p\} \subset \mathbb{R}^n$, sparse coding is about determining a *dictionary* (the set of atoms)

$$\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_m\} \subset \mathbb{R}^n,$$

together with a set of coefficient vectors $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_p\} \subset \mathbb{R}^m$ with most elements close to zero, so that each input vector $y_j$ can be approximated by the linear combination $\boldsymbol{y}_j \approx \sum_{\ell=1}^{m} \boldsymbol{c}_j(\ell)\boldsymbol{d}_\ell$. The typical sparse coding method, e.g. K-SVD [1], determines the dictionary $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_m\}$ via solving an optimization problem with sparsity-prompting functional on the coefficients:

$$\min_{\boldsymbol{D}, \{\boldsymbol{c}_i\}_{i=1}^{p}} \sum_{i=1}^{p} (\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{c}_i\|_2^2 + \lambda \|\boldsymbol{c}_i\|_0), \quad \text{subject to } \|\boldsymbol{d}_j\|_2 = 1, \ 1 \le j \le m, \quad (1)$$

where $\|\cdot\|_0$ counts the number of non-zero entries and $\boldsymbol{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m\}$ is the dictionary for sparse coding. It is well known that the above minimization

(1) is an NP-hard problem and only sub-optimal solution can be obtained in polynomial time. Most existing methods use an alternating iteration scheme to solve (1).

Despite the success of sparse coding in many applications, the sequence generated by most existing numerical solvers for solving the non-convex problem (1) can only guarantee that the functional value of (1) is decreasing at each iteration, which can not guarantee the generated sequence is convergent. Indeed, the sequence generated by the K-SVD method is not convergent; see Fig. 1 for an illustration. Moreover, as it has been mentioned in the literature, good performance of sparse coding in various recognition tasks requires imposing some additional constraints of the dictionary. One of such essential dictionary properties is the so-called *mutual coherence*:

$$\mu(\boldsymbol{D}) = \max_{i \neq j} |\langle \boldsymbol{d}_i, \boldsymbol{d}_j \rangle|, \tag{2}$$

which further increases the technical difficulty of designing an effective numerical method with theoretical soundness. Although there is no such term in (1), the existing implementation of the K-SVD method implicitly tries to avoid learning a dictionary with high mutual coherence by discarding the learned atom which has large mutual coherence with the existing ones in each iteration.

In this paper, we consider the problem of sparse coding that explicitly imposes additional regularization on the mutual coherence of the dictionary, which can be formulated as the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{D}, \{\boldsymbol{c}_i\}_{i=1}^{p}} \quad & \sum_i (\frac{1}{2} \|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{c}_i\|_F^2 + \lambda \|\boldsymbol{c}_i\|_0) + \frac{\alpha}{2} \|\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{I}\|_F^2, \\ s.t. \quad & \|\boldsymbol{d}_j\|_2 = 1, \ 1 \leq j \leq m. \end{aligned} \tag{3}$$

The minimization models similar to (3) have been used in several sparse coding based systems; see e.g. [21, 16, 7]. As a more general optimization problem which contains the K-SVD model (1) by setting $\alpha = 0$, the optimization problem (3) is a even harder problem to solve.

This paper aims at developing a fast alternating iteration scheme specifically designed for solving (3). As shown in the experiments, compared to the generic dictionary generated by the K-SVD method, the dictionary generated by the proposed method has much lower mutual coherence and it provides better performance in several sparse coding based recognition tasks. Moreover, in contrast to the existing numerical solvers for (3), we provided the rigorous analysis on the convergence of the proposed method. It is mathematically proved that the whole sequence generated by the proposed method converges to a stationary point of the problem, while the existing analysis of all other solvers only shows that the functional values of the sequence is decreasing or equivalently only a sub-sequence is convergent. The whole sequence convergence of an iteration scheme is not only of theoretical interest, but also important for applications, e.g. the number of iterations does not need to be empirically chosen for obtaining stability.

## 1.1 Motivation and Main Contributions

The main motivation of this paper is two-fold: one is the need for learning an incoherent dictionary for sparse coding in many applications, and the other is the need of a numerical solver for solving (3) with proved convergence property.

**Motivation** *The need of an incoherent dictionary for sparse coding.* Once a dictionary is learned, the sparse code for each input is then computed via some pursuit methods, e.g. *orthogonal matching pursuit* [25], *basis pursuit* [10]. The success of these methods for finding the optimal sparse code depends on the incoherence property of the dictionary. In [25], Tropp showed that the OMP can recover the exact support of the coefficients whenever mutual coherence $\mu$ is less that $1/(2S-1)$ where $S$ is the number of nonzero entries of the correct coefficients. It is further proved in [23] that the similar requirement on the mutual coherence is also needed for ensuring the correctness of the thresholding-based sparse coding algorithms. In practice, it is also observed that a dictionary with high mutual coherence will impact the performance of sparse coding based methods; see e.g [21, 26, 8].
*The need of a variational model that explicitly regularizes mutual coherence.* In a quick glance, the widely used K-SVD method [1] for sparse coding considered a variational model which has no explicit functional on minimizing the mutual coherence of the result, i.e., it considered a special case of (3) with $\alpha = 0$. However, the implementation of the K-SVD method implicitly controlled the mutual coherence of the dictionary by discarding the "bad" atom which is highly correlated to the ones already in the dictionary. Such an ad-hoc approach certainly is not optimal for lowering the overall mutual coherence of the dictionary. In practice, the K-SVD method may still give a dictionary that contains highly correlated atoms, which will lead to poor performance in sparse approximation, see [11] for more details.
*The need of a convergent algorithm.* The minimization problem (3) is a challenging non-convex problem. Most existing methods that used the model (3) or its extensions, e.g. [15, 28, 18], simply call some generic non-linear optimization solvers such as the *projected gradient* method. Such a scheme is slow and not stable in practice. Furthermore, all these methods at most can be proved that the functional value is decreasing at each iteration. The sequence itself may not be convergent. From the theoretical perspective, a non-convergent algorithm certainly is not satisfactory. From the application perspective, the divergence of the algorithm also leads to troublesome issues such as when to stop the numerical solver, which often requires manual tune-up.

**Main Contributions** In this paper, we proposed a hybrid alternating proximal scheme for solving (3). Compared to the K-SVD method that controls the mutual coherence of the dictionary in an ad-hoc manner, the proposed method is optimized for learning an incoherent dictionary for sparse coding. Compared to the generic numerical scheme for solving (3) adopted in the existing applications, the convergence property of the proposed method is rigorously established in the paper. We showed that the whole sequence generated by the proposed method converges to a stationary point. As a comparison, only sub-sequence convergence can be proved for existing numerical methods. The whole sequence convergence of an iteration scheme is not only of theoretical interest, but also important for applications as the number of iterations does not need to be empirically chosen to keep the output stable.

### 1.2 Related Work

In this section, we gives a brief review on most related generic dictionary learning methods and incoherent dictionary learning methods for sparse coding.

**Generic Dictionary Learning Methods** Among many existing dictionary learning methods, the so-called K-SVD method [1] is the most widely used one. The K-SVD method solves the problem (3) with $\alpha = 0$ by alternatively iterating between sparse code $\boldsymbol{C}$ and the dictionary $\boldsymbol{D}$. The sparse code $\boldsymbol{C}$ is estimated by using the OMP method [25]: at each step, one atom is selected such that it is most correlated with the current residuals and finally the observation is projected onto the linear space spanned by the chosen atoms. In the dictionary update stage for estimating $\boldsymbol{D}$, the atoms are updated sequentially by using the rank-1 approximation to current residuals which can be exactly solved by the SVD decomposition. Most other existing dictionary learning methods (e.g. [18, 17, 2, 14]) are also based on the similar alternating scheme between the dictionary update and sparse code estimation. In [17, 14], the atoms in the dictionary are updated sequentially with closed form solutions. The projection gradient descent method is used in [18] to update the whole dictionary. For the $\ell_0$ norm related minimization problem in the stage of sparse code estimation, many relaxation methods have been proposed and the $\ell_1$ norm based relaxation is the most popular one; see e.g. [18, 17, 14, 27]. Among these methods, the convergence analysis is provided in [27] for its proximal method. Recently, an proximal alternating linearized method is presented in [6] to directly solve the $\ell_0$ norm based optimization problem for dictionary learning. The method proposed in [6] is mathematically proven to be globally convergent.

**Incoherent Dictionary Learning Methods** There are two types of approaches to learn an incoherent dictionary for sparse coding. The first one is to add an additional process in the existing generic dictionary learning method to lower the mutual coherence, e.g. [16, 7]. Both [16] and [7] added the decorrelation step after the dictionary update stage in K-SVD method. In [16], the de-correlation is done via minimizing the distance between the learned dictionary generated by the K-SVD method and the space spanned by the dictionaries with certain mutual coherence level. However, this projection step doesn't consider the approximation error and may significantly increase the whole minimization functional value. Thus, in [7], the iterative projection method is introduced to lower the mutual coherence of the dictionary, together with an additional dictionary rotation step to improve the approximation error of the de-correlated dictionary. The other way to learn the incoherent dictionary is directly solving a minimization model that contains the functional related the mutual coherence of the dictionary, e.g. [21, 5]. In [21], an additional regularization term on mutual coherence is added to (1) when being applied in image classification and clustering. The approach presented in [7] used the OMP method in sparse code estimation and method of optimal coherence-constrained direction for dictionary update. In [5], the orthogonality constraints on the dictionary atoms are explicitly added in the variational model for dictionary learning such that its mutual coherence is always 0. With the performance comparable to the K-SVD method in image recovery, the orthogonal dictionary based method [5] is significantly faster than the K-SVD method. Such advantages on computational efficiency comes from the fact that both sparse code estimation and dictionary update have closed-form solutions in [5].

## 2 Incoherent Dictionary Learning Algorithm

We first give an introduction to the definitions and notations used in this section. We define $\boldsymbol{Y}$ be a matrix, $\boldsymbol{y}_j$ be the $j-$th column of $\boldsymbol{Y}$ and $y_{ij}$ be the $(i,j)-$th element of $\boldsymbol{Y}$. Given the matrix $\boldsymbol{Y}$, the Frobenius norm of $\boldsymbol{Y}$ is defined by $\|\boldsymbol{Y}\|_F = (\sum_{i,j} y_{ij}^2)^{1/2}$, its $\ell_0$ norm $\|\boldsymbol{Y}\|_0$ is defined as the number of nonzero entries of $\boldsymbol{Y}$ and the infinity norm of $\|\boldsymbol{Y}\|_\infty = \max_{i,j}\{|y_{ij}|\}$. Define the *hard thresholding operator* $T_\lambda(\boldsymbol{D})[i,j] = d_{ij}$ if $|d_{ij}| > \lambda$ and $T_\lambda(\boldsymbol{D})[i,j] = 0$ otherwise.

### 2.1 Problem Formulation

Given the training samples $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p) \in \mathbb{R}^{n\times p}$, we consider the sparse approximation of $\boldsymbol{Y}$ by the redundant dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times m}$. Same as [21], we can introduce the regularization $\|\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{I}\|_F^2$ to the variational model to minimize the mutual coherence. The variational model of incoherent dictionary learning model is given as follows,

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{C}} \quad & \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_0 + \frac{\alpha}{2}\|\boldsymbol{D}^\top\boldsymbol{D} - \boldsymbol{I}\|_F^2, \\ s.t. \quad & \|\boldsymbol{d}_j\|_2 = 1,\ 1 \le j \le m; \|\boldsymbol{c}_i\|_\infty \le M,\ 1 \le i \le m, \end{aligned} \tag{4}$$

where $\boldsymbol{D} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m) \in \mathbb{R}^{n\times m}$, $\boldsymbol{C} = (\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_m^\top)^\top \in \mathbb{R}^{m\times p}$ and $M$ is the predefined upper bound for the elements in $\boldsymbol{C}$. It is noted that the predefined upper bound $M$ is mainly for the stability of the algorithm, which is allowed to be set arbitrarily large. For the simplicity of discussion, define $\mathcal{D} = \{\boldsymbol{D} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m) \in \mathbb{R}^{n\times m} : \|\boldsymbol{d}_j\|_2 = 1,\ 1 \le j \le m\}$ and $\mathcal{C} = \{\boldsymbol{C} = (\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_m^\top)^\top \in \mathbb{R}^{m\times p}, \|\boldsymbol{c}_i\|_\infty \le M,\ 1 \le i \le m\}$. Then the model (4) can be reformulated as

$$\min_{\boldsymbol{D},\boldsymbol{C}} \quad \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_0 + \frac{\alpha}{2}\|\boldsymbol{D}^\top\boldsymbol{D} - \boldsymbol{I}\|_F^2, \text{ s.t. } \boldsymbol{D} \in \mathcal{D},\ \boldsymbol{C} \in \mathcal{C}. \tag{5}$$

In the next, we will propose the hybrid alternating proximal algorithm for solving (5) with the whole sequence convergence property.

## 2.2 A Hybrid Alternating Proximal Algorithm

The algorithm for solving (4) is based on a hybrid scheme that combines the alternating proximal method [3] and the alternating proximal linearized method [9], which are about tackling the non-convex minimization problem of the form:

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

where $F, G$ are proper lower semi-continuous functions and $Q$ is the smooth function with Lipschitz derivatives on any bounded set, that is, for the bounded set $\mathcal{Z}$, there exists a constant $L > 0$, such that $\|\nabla Q(z_1) - \nabla Q(z_2)\|_F \leq L\|z_1 - z_2\|_F, z_1, z_2 \in \mathcal{Z}$.

The alternating proximal method [3] updates the $(x, y)$ via as follows,

$$\begin{cases} x_{k+1} \in \arg\min_x F(x) + Q(x, y_k) + G(y_k) + \frac{\mu^k}{2}\|x - x_k\|_F^2; \\ y_{k+1} \in \arg\min_x F(x_{k+1}) + Q(x_{k+1}, y) + G(y) + \frac{\lambda^k}{2}\|y - y_k\|_F^2, \end{cases} \tag{7}$$

where $\mu^k, \lambda^k$ are suitable step sizes. In general, the scheme (7) requires solving the non-smooth and non-convex minimization problems in each step which often has no closed form solutions. This motivates a linearized version of alternating proximal algorithm [9] such that each subproblem has a closed form solution. Instead of solving the subproblems as (7), the alternating proximal linearized algorithm replaces the smooth term $Q$ in (7) by its first order linear approximation:

$$\begin{cases} x_{k+1} \in \arg\min_x F(x) + \hat{Q}_{(x_k,y_k)}(x) + G(y_k) + \frac{\mu^k}{2}\|x - x_k\|_F^2; \\ y_{k+1} \in \arg\min_y F(x_{k+1}) + \hat{Q}_{(x_{k+1},y_k)}(y) + G(y) + \frac{\lambda^k}{2}\|y - y_k\|_F^2. \end{cases} \tag{8}$$

where $\hat{Q}_{(x_k,y_k)}(x) = Q(x_k, y_k) + \langle\nabla_x Q(x_k, y_k), x - x_k\rangle$, $\hat{Q}_{(x_k,y_k)}(y) = Q(x_k, y_k) + \langle\nabla_y Q(x_k, y_k), y - y_k\rangle$, and $\mu^k, \lambda^k$ are carefully chosen step sizes.

Although the proximal linearized method has closed form solutions for all sub-problems, it requires more iterations to converge than the proximal method as it only provides approximated solutions to two-subproblems in (7). The problem (5) we are solving is different from the generic model considered in the proximal method, as the first sub-problem for sparse code estimation in (7) has a closed-form solution while the second one does not. Motivated by this observation, we proposed a hybrid iteration scheme which uses the formulation of the proximal method for sparse code estimation and uses the formulation of the proximal linearized method for dictionary update. In other words, it is a hybrid version that combines both the proximal method and the proximal linearized method. As a result, the proposed one also has the closed form solutions for all sub-problems at each iteration, but converges faster than the proximal linearized method.

*Remark 1.* Although both (7) and (8) are the alternating schemes between two variables, they can be extended to the case of the alternating iteration among a finite number of blocks [9, 4].

The iterations (7) and (8) can be re-written by using the *proximal operator* [22]:

$$\mathrm{P}rox_t^F(x) := \arg\min_u F(u) + \frac{t}{2}\|u - x\|_F^2.$$

Then, the minimization (7) can be re-written as

$$\begin{cases} x_{k+1} \in \mathrm{P}rox_{\mu^k}^{F+Q(\cdot,y_k)}(x_k), \\ y_{k+1} \in \mathrm{P}rox_{\lambda^k}^{G+Q(x_{k+1},\cdot)}(y_k), \end{cases} \tag{9}$$

and the minimization (8) can be re-written as

$$\begin{cases} x_{k+1} \in \mathrm{P}rox_{\mu^k}^{F}(x_k - \frac{1}{\mu^k}\nabla_x Q(x_k, y_k)), \\ y_{k+1} \in \mathrm{P}rox_{\lambda^k}^{G}(y_k - \frac{1}{\lambda^k}\nabla_y Q(x_{k+1}, y_k)). \end{cases} \tag{10}$$

*Remark 2.* It is shown in [9] that the proximal operator defined in (9), (10) are well defined, i.e., the solution sets of (7) and (8) are nonempty and compact.

The minimization (4) can be expressed in the form (6) by setting

$$\begin{cases} F(\boldsymbol{C}) = \lambda\|\boldsymbol{C}\|_0 + \delta_{\mathcal{C}}(\boldsymbol{C}), \\ Q(\boldsymbol{C}, \boldsymbol{D}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{D}^\top\boldsymbol{D} - \boldsymbol{I}\|_F^2, \\ G(\boldsymbol{D}) = \delta_{\mathcal{D}}(\boldsymbol{D}), \end{cases} \tag{11}$$

where $\delta_{\mathcal{C}}(\boldsymbol{C})$ and $\delta_{\mathcal{D}}(\boldsymbol{D})$ are indicator functions, that is $\delta_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$ and $\delta_{\mathcal{X}}(x) = +\infty$ if $x \notin \mathcal{X}$. We propose the following alternating scheme to solve (4).

**Sparse Code Estimator** given the dictionary $\boldsymbol{d}^{(k)}$, we update the sparse code $\boldsymbol{c}^{(k)} = \{\boldsymbol{c}_j^\top\}_{j=1}^m$ row by row as follows:

$$\boldsymbol{c}_j^{(k)} \in \mathrm{P}rox_{\mu_j^k}^{F(\boldsymbol{U}_j^k)+Q(\boldsymbol{U}_j^k,\boldsymbol{D}^{(k)})}(\boldsymbol{c}_j^{(k-1)}), \quad 1 \leq j \leq m, \tag{12}$$

where $\boldsymbol{U}_j^k = (\boldsymbol{c}_1^{(k)\top}, \ldots, \boldsymbol{c}_{j-1}^{(k)\top}, \boldsymbol{c}_j^\top, \boldsymbol{c}_{j+1}^{(k-1)\top}, \ldots, \boldsymbol{c}_m^{(k-1)\top})^\top$ for $1 \leq j \leq m$. The minimization (12) is easy to solve as it has closed form solution. Define $\mathcal{S}_j^k = \{i | d_{ij} \neq 0, 1 \leq i \leq n\}$ and $\boldsymbol{R}^{j,k} = \boldsymbol{Y} - \sum\limits_{i<j} \boldsymbol{d}_i^{(k)}\boldsymbol{c}_i^{(k)} - \sum\limits_{i>j} \boldsymbol{d}_i^{(k)}\boldsymbol{c}_i^{(k-1)}$. By direct calculation, the minimization (12) is equivalent to

$$\boldsymbol{c}_j^{(k)} \in \arg\min_{\boldsymbol{c}_j \in \mathcal{C}} \frac{\mu_j^k}{2}\|\boldsymbol{c}_j - \boldsymbol{c}_j^{(k-1)}\|_F^2 + \frac{1}{2}\sum_{i \in \mathcal{S}_j^k} \|\boldsymbol{r}_i^{j,k} - d_{ij}\boldsymbol{c}_j\|_F^2 + \lambda\|\boldsymbol{c}_j\|_0, \tag{13}$$

where $\boldsymbol{R}^{j,k} = (\boldsymbol{r}_1^{j,k\top}, \ldots, \boldsymbol{r}_n^{j,k\top})^\top \in \mathbb{R}^{n\times p}$.

**Proposition 1.** *Suppose $M$ is chosen such that $M > \sqrt{\frac{2\lambda}{r_j^k}}$, where $r_j^k = \sum_{i\in\mathcal{S}_j^k} d_{ij}^2 + \mu_j^k$, the minimization* (13) *has the closed form solution for all* $1 \leq j \leq m$*, given by*

$$\boldsymbol{c}_j^{(k)} = \min(T_{\sqrt{2\lambda/r_j^k}}((\sum_{i\in\mathcal{S}_j^k} d_{ij}\boldsymbol{r}_i^{j,k} + \mu_j^k \boldsymbol{c}_j^{(k-1)})/r_j^k), M). \tag{14}$$

*Proof.* By direct calculation, it can be seen that he minimization (13) is equivalent to the following minimization.

$$\boldsymbol{c}_j^{(k)} \in \operatorname*{arg\,min}_{\boldsymbol{c}_j\in\mathcal{C}} r_j^k \|\boldsymbol{c}_j - (\sum_{i\in\mathcal{S}_j^k} d_{ij}\boldsymbol{r}_i^{j,k} + \mu_j^k \boldsymbol{c}_j^{(k-1)})/r_j^k\|_F^2 + 2\lambda\|\boldsymbol{c}_j\|_0. \tag{15}$$

The variables in the minimization (15) above are separable. Thus, it is easy to see that the solution of (15) is exactly the one defined by (14).

**Dictionary Update** Given the sparse code $\boldsymbol{c}^{(k)}$, we update the dictionary $\boldsymbol{D}^{(k+1)} = \{\boldsymbol{d}_j\}_{j=1}^m$ atom by atom as follows:

$$\boldsymbol{d}_j^{(k+1)} \in \mathrm{P}rox_{\lambda_j^k}^{G(\boldsymbol{S}_j^{(k)})}(\boldsymbol{d}_j^{(k)} - \frac{1}{\lambda_j^k}\nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \boldsymbol{V}_j^k)), \tag{16}$$

where

$$\begin{cases} \boldsymbol{S}_j^k = (\boldsymbol{d}_1^{(k+1)}, \ldots, \boldsymbol{d}_{j-1}^{(k+1)}, \boldsymbol{d}_j, \boldsymbol{d}_{j+1}^{(k)}, \ldots, \boldsymbol{d}_m^{(k)}), \\ \boldsymbol{V}_j^k = (\boldsymbol{d}_1^{(k+1)}, \ldots, \boldsymbol{d}_{j-1}^{(k+1)}, \boldsymbol{d}_j^{(k)}, \boldsymbol{d}_{j+1}^{(k)}, \ldots, \boldsymbol{d}_m^{(k)}). \end{cases}$$

Denote $\boldsymbol{d}^{j,k} = \boldsymbol{d}_j^{(k)} - \frac{1}{\lambda_j^k}\nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \boldsymbol{V}_j^k)$, Then (16) can be reformulated as:

$$\boldsymbol{d}_j^{(k+1)} \in \operatorname*{arg\,min}_{\|\boldsymbol{d}_j\|_2=1} \|\boldsymbol{d}_j - \boldsymbol{d}^{j,k}\|_2^2, \tag{17}$$

From (17), it is easy to know $\boldsymbol{d}_j^{(k+1)} = \boldsymbol{d}^{j,k}/\|\boldsymbol{d}^{j,k}\|_2$ for $1 \leq j \leq m$.

There are two step sizes, $\mu_j^k$ and $\lambda_j^k$ needed to be set in the calculation. The step size $\mu_j^k$ can be set arbitrarily as long as there exists $a, b > 0$ such that $\mu_j^k \in (a, b)$, $\forall k = 1, 2, \ldots, j = 1, \ldots, m$. The step size $\lambda_j^k$ can be chosen as $\lambda_j^k = \max(a, \rho L(\boldsymbol{d}_j^{(k)}))$, where the $\lambda_j^k$ can be chosen so as to

$$\|\nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \bar{\boldsymbol{D}}_j^1) - \nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \bar{\boldsymbol{D}}_j^2)\|_F \leq L(\boldsymbol{d}_j^k)\|\boldsymbol{d}_j^1 - \boldsymbol{d}_j^2\|_F,$$

for all $\boldsymbol{d}_j^1, \boldsymbol{d}_j^2 \in \mathbb{R}^n$ where $\bar{\boldsymbol{D}}_j^i = (\boldsymbol{d}_1^{(k+1)}, \ldots, \boldsymbol{d}_{j-1}^{(k+1)}, \boldsymbol{d}_j^i, \boldsymbol{d}_{j+1}^{(k)}, \ldots, \boldsymbol{d}_m^{(k)}), i = 1, 2$. Typically, we can choose $\mu_j^k = \mu_0$ and $L(\boldsymbol{d}_j^k) = \boldsymbol{c}_j^{(k)} \boldsymbol{c}_j^{(k)^\top} + \alpha\|\boldsymbol{V}_j^k\|_2$ for all $j = 1, 2, \ldots, m$ and $k = 1, 2, \ldots$. It can been seen that $L(\boldsymbol{d}_j^k)$ is a bounded sequence since $\boldsymbol{C}$ is bounded in the model (5). See the Alg. 1 for the outline of the proposed incoherent dictionary learning method that solves (5).

**Algorithm 1** Incoherent dictionary learning algorithm via solving (5).

```
INPUT: Training signals $\boldsymbol{Y}$;
OUTPUT: Learned Incoherent Dictionary $\boldsymbol{D}$;
Main Procedure:
    1. Set the initial dictionary $\boldsymbol{D}^{(0)}$, $\rho > 1$, $a > 0$ and $K \in \mathbb{N}$.
    2. For $k = 0, 1, \ldots, K$,
     (a) Sparse Coding: for $j = 1, \ldots, m$, let $\mathcal{S}_j^k = \{i : d_{ij}^{(k)} \neq 0, 1 \leq i \leq n\}$,

         $\boldsymbol{r}^{j,k} = \boldsymbol{Y} - \sum_{i<j} \boldsymbol{d}_i^{(k)} \boldsymbol{c}_i^{(k)} - \sum_{i>j} \boldsymbol{d}_i^{(k)} \boldsymbol{c}_i^{(k-1)},$
         $\boldsymbol{c}^{j,k} = \sum_{i \in \mathcal{S}_j^k} d_{ij} \boldsymbol{r}_i^{j,k} + \mu_j^k \boldsymbol{c}_j^{(k-1)}, \; r_j^k = \sum_{i \in \mathcal{S}_j^k} d_{ij}^2 + \mu_j^k,$   (18)
         $\boldsymbol{c}_j^{(k)} = \min(T_{\sqrt{2\lambda / r_j^k}}(\boldsymbol{c}^{j,k} / r_j^k), M).$

     (b) Update the step size: for $j = 1, \ldots, m$

         $V^{(k)} = \boldsymbol{C}^{(k)} \boldsymbol{C}^{(k)\top}, \quad L(\boldsymbol{d}_j^{(k)}) = V_{j,j}^{(k)} + \alpha \|\boldsymbol{V}^k\|_2.$

     (c) Dictionary Update: let $\mu_j^k = \max\{\rho L(\boldsymbol{d}_j^k), a\}$, for $k = 1, \ldots, m$,

         $\boldsymbol{d}^{j,k} = \boldsymbol{d}_j^{(k)} - \frac{1}{\mu_l^k} \nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \boldsymbol{V}_j^k); \qquad \boldsymbol{d}_j^{(k+1)} = \boldsymbol{d}^{j,k} / \|\boldsymbol{d}^{j,k}\|_2.$   (19)
```

## 3 Convergence Analysis of Algorithm 1

Before proving the convergence property of the Alg.1, we define the critical points for the non-convex and non-smooth functions [9].

**Definition 1.** *Given the non-convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper and lower semi-continuous function and* $\mathrm{dom} f = \{x \in \mathbb{R}^n : f(x) < +\infty\}$.

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

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

  *and $\hat{\partial} f(x) = \emptyset$ if $x \notin \mathrm{dom} f$.*
- *The* Limiting Subdifferential *of $f$ at $x$ is defined as*

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

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

*Remark 3.* (i) If $x$ is a local minimizer of $f$ then $0 \in \partial f(x)$. (ii) If $f$ is the convex function, then $\partial f(x) = \hat{\partial} f(x) = \{u | f(y) \geq f(x) + \langle u, y - x \rangle, \forall y \in \mathrm{dom} f\}$. In that case, $0 \in \partial f(x)$ is the first order optimal condition.

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

*Proof.* See Appendix A.

## 4 Experiments

We used the proposed incoherent dictionary learning method in sparse coding based recognition systems. The basic procedure is as follows. Firstly, the dictionary is learned from the training set using Alg. 1. Then, the sparse code $C$ for each sample in the training set, as well as the test set, is calculated using the proximal alternating algorithm [20]. At last, a linear classifier is trained and tested on the sparse codes. Two applications are considered in the experiments: face recognition and object classification. The experimental results showed that using the incoherent dictionary learned from the proposed method, the sparse coding based recognition systems may have some additional performance gain.

### 4.1 Experimental Setting

The performance is evaluated on two applications: face recognition on the Extended YaleB dataset [13] and the AR face dataset [19], and object classification on the Caltech-101 dataset [12]. Our approach is compared to two dictionary learning based methods:

- *K-SVD (Baseline)* [1] : The basic procedure is similar to ours, i.e., the dictionary is trained using K-SVD and the sparse codes are used to train a linear classifier. The dictionary learning process and the classifier training process are independent.
- *D-KSVD* [28] : This method is an extension of the above baseline method, which incorporates the classification error into the objective function of K-SVD dictionary learning. The dictionary and the linear classifier are trained simultaneously.

Note that both methods are built upon the K-SVD dictionary learning method [1] which does not impose dictionary incoherence, and all the tested methods are based on a simple linear classifier. The experimental setting is as follows:

- *Extended Yale B* : The extended YaleB database [13] 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. Each face image is projected 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 was used for training the dictionary which contains 32 images for each person, and the other half was used for testing.

- *AR Face* : The AR face database [19] 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. A subset of the database consisting of 2,600 images from 50 male subjects and 50 female subjects is used. 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.
- *Caltech101* : The Caltech101 dataset [12] contains $9,144$ images from 102 classes (i.e., 101 object categories with 8677 images and one additional background category with 467 images) including vehicles, plants, animals, cartoon characters, and so on. The number of images in each category varies from 31 to 800. We use 20 samples per category for training the dictionary as well as the classifier and the rest for testing. The spatial pyramid feature presented in [28] is computed on each image as input.

To obtain reliable results, each experiment is repeated 30 times with different random splits of the training and testing images. The final classification accuracies are reported as the average of each run. Throughout the experiments, we fix the sparsity parameter $\lambda$ to be 0.005 and the coherence parameter $\beta$ to be 1. The iteration number $K$ in Alg. 1 is fixed to be 10. The dictionary size is set 540 on the two face datasets and 3000 on the Caltech-101 dataset.

### 4.2 Experimental Results

The results and the conclusions are summarized as follows.

- **Convergence behavior.** The convergence behaviors of the K-SVD method and Alg. 1 on the YaleB face dataset are compared in Fig. 1, which plots the Frobenius norm of the increments of the sparse codes generated by two algorithms at each iteration. It can be seen that the code sequence generated by the K-SVD method does not converge to zero, which means that the K-SVD method has at most sub-sequence convergence. In contrast, the increments of the code sequence generated by Alg. 1 converges to zero which shows that the whole sequence converges.
- **Mutual coherence of dictionary.** The matrices of the mutual coherence of the dictionaries learned from the YaleB dataset are shown in Fig. 3, and its normalized histograms are shown in Fig. 2. It can be seen that mutual coherence of the dictionary from our approach can be significantly lower than that from the K-SVD method when the regularization parameter $\beta$ on mutual coherence is set sufficiently large.
- **Classification performance.** The classification results are listed in Table 1. It can be seen that our approach performs slightly better than the compared methods.

**Table 1.** Classification accuracies (%) on two face datasets and one object dataset.

| Dataset | K-SVD | D-KSVD | Ours |
|---|---|---|---|
| Extended YaleB | 93.10 | 94.10 | 95.72 |
| AR Face | 86.50 | 88.80 | 96.18 |
| Caltech-101 | 68.70 | 68.60 | 72.29 |

## 5 Summary and Conclusions

This paper aims at developing an alternating iteration scheme for learning an incoherent dictionary, which is the first available incoherent dictionary learning method with proved sequence convergence. The proposed work not only is of theoretical interest from the viewpoint of optimization, but also might be useful to practical sparse coding based applications.

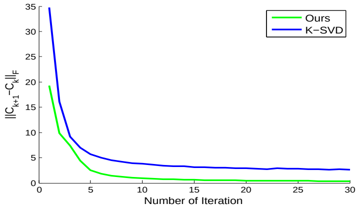


**Fig. 1.** The increments of the sequences generated by the methods.

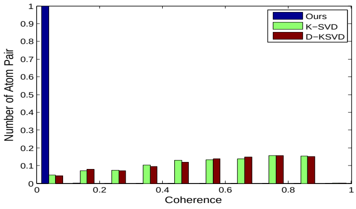
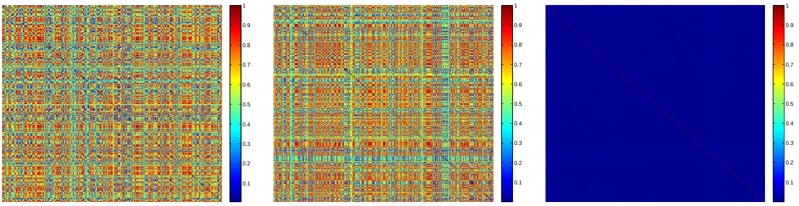


**Fig. 2.** The normalized histograms on the coherence matrices shown in Fig. 3.



(a) K-SVD (b) D-KSVD (c) Alg.1

**Fig. 3.** The mutual coherence matrices of the dictionaries learned from the YaleB face dataset using the K-SVD method and Alg.1. The $i$th-column and $j$th-row element in each matrix represents the mutual coherence between the $i$th and $j$-th atom.

## 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.

## Appendix A

In this appendix, we give a sketch of the proof of Theorem 1. The detailed proof is provided in the complementary material. The proof of Theorem 1 is built upon Theorem 2.9 in [4].

**Theorem 2.** *([4]) Assume $H(z)$ is a proper and lower semi-continuous function with* $\inf H > -\infty$*, the sequence $\{z^{(k)}\}_{k\in\mathbb{N}}$ is a Cauchy sequence and converges to the critical point of $H(z)$, if the following four conditions hold:*

(P1) **Sufficient decrease condition.** *There exists some positive constant $\rho_1$, such that*

$$H(z^{(k)}) - H(z^{(k+1)}) \geq \rho_1 \|z^{(k+1)} - z^{(k)}\|_F^2, \ \forall k = 1, 2, \ldots.$$

(P2) **Relative error condition.** *There exists some positive constant $\rho_2 > 0$, such that*

$$\|w^{(k+1)}\|_F \leq \rho_2 \|z^{(k+1)} - z^{(k)}\|_F, \ w^{(k)} \in \partial H(z^{(k)}), \ \forall k = 1, 2, \ldots.$$

(P3) **Continuity condition.** *There exists a subsequence $\{z^{(k_j)}\}_{j\in\mathbb{N}}$ and $\bar{z}$ such that*

$$z^{(k_j)} \to \bar{z}, \ H(z^{(k_j)}) \to H(\bar{z}), \quad \text{as } j \to +\infty.$$

(P4) $H(z)$ **is a KL function**. *$H(z)$ satisfies the* Kurdyka-Lojasiewicz *property in its effective domain.*

Let $\boldsymbol{Z}^{(k)} := (\boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)})$ denote the sequence generated by the algorithm 1. Firstly, it can be seen that the objective function $H(\boldsymbol{Z}) = F(\boldsymbol{C}) + Q(\boldsymbol{Z}) + G(\boldsymbol{D})$ is the proper, lower semi-continuous function and bounded below by 0 where $F, Q, G$ are defined in (11). Secondly, the sequence $\{\boldsymbol{Z}^{(k)}\}_{k\in\mathbb{N}}$ generated by algorithm 1 is bounded since $\boldsymbol{D}^{(k)} \in \mathcal{D}$ and $\boldsymbol{C}^{(k)} \in \mathcal{C}$ for all $k = 1, 2, \ldots$. In the next, we show that the sequence $\{\boldsymbol{Z}^{(k)}\}$ satisfies the condition (P1)-(P4) using the following four lemmas. The proofs of these lemmas are presented in supplemental materials.

**Lemma 1.** *The sequence $\{\boldsymbol{Z}^{(k)}\}_{k\in\mathbb{N}}$ satisfies*

$$\begin{cases} H(\boldsymbol{T}_j^{(k+1)}, \boldsymbol{D}^{(k)}) \leq H(\boldsymbol{T}_{j-1}^{(k+1)}, \boldsymbol{D}^{(k)}) - \frac{\mu_j^k}{2} \|\boldsymbol{c}_j^{(k+1)} - \boldsymbol{c}_j^{(k)}\|_F^2, \\ H(\boldsymbol{C}^{(k+1)}, \boldsymbol{V}_j^{(k+1)}) \leq H(\boldsymbol{C}^{(k+1)}, \boldsymbol{V}_{j-1}^{(k+1)}) - \frac{\lambda_j^k - L(\boldsymbol{d}_j^{(k)})}{2} \|\boldsymbol{d}_j^{(k+1)} - \boldsymbol{d}_j^{(k)}\|_F^2, \end{cases}$$

*for $1 \leq j \leq m$, where*

$$\begin{cases} \boldsymbol{T}_j^{(k)} = (\boldsymbol{c}_1^{(k)\top}, \ldots, \boldsymbol{c}_j^{(k)\top}, \boldsymbol{c}_{j+1}^{(k-1)\top}, \ldots, \boldsymbol{c}_m^{(k-1)\top})^\top, \quad \boldsymbol{T}_0^{(k)} = \boldsymbol{C}^{(k-1)}, \\ \boldsymbol{V}_j^{(k)} = (\boldsymbol{d}_1^{(k)}, \ldots, \boldsymbol{d}_j^{(k)}, \boldsymbol{d}_{j+1}^{(k-1)}, \ldots, \boldsymbol{d}_m^{(k-1)}), \quad \boldsymbol{V}_0^{(k)} = \boldsymbol{D}^{(k-1)}. \end{cases} \tag{20}$$

Sum up the above inequalities, we can obtain

$$\begin{aligned} & H(\boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}) - H(\boldsymbol{C}^{(k+1)}, \boldsymbol{D}^{(k+1)}) \\ \geq & \sum_{j=1}^{m} (\frac{\mu_j^k}{2} \|\boldsymbol{c}_j^{(k+1)} - \boldsymbol{c}_j^{(k)}\|_F^2 + \frac{\lambda_j^k - L(\boldsymbol{d}_j^{(k)})}{2} \|\boldsymbol{d}_j^{(k+1)} - \boldsymbol{d}_j^{(k)}\|_F^2). \end{aligned} \tag{21}$$

Using the fact that there exist $a, b > 0$ such that $a < \mu_j^k, \lambda_j^k < b$ and $\lambda_j^k > L(\boldsymbol{d}_j^{(k)})$, we can establish the sufficient decreasing property (P1) for $\{\boldsymbol{Z}^{(k)}\}_{k \in \mathbb{N}}$.

**Lemma 2.** *Let $\boldsymbol{\omega}_{\boldsymbol{C}}^{(k)} = (\boldsymbol{\omega}_{\boldsymbol{C}}^{1\top}, \ldots, \boldsymbol{\omega}_{\boldsymbol{C}}^{m\top})^\top$ and $\boldsymbol{\omega}_{\boldsymbol{D}}^{(k)} = (\boldsymbol{\omega}_{\boldsymbol{D}}^{1}, \ldots, \boldsymbol{\omega}_{\boldsymbol{D}}^{m})$ where*

$$\begin{cases} \boldsymbol{\omega}_{\boldsymbol{C}}^{j} = \nabla_{\boldsymbol{c}_j} Q(\boldsymbol{Z}^{(k)}) - \nabla_{\boldsymbol{c}_j} Q(\boldsymbol{T}_j^{(k)}, \boldsymbol{D}^{(k-1)}) - \mu_j^k(\boldsymbol{c}_j^{(k)} - \boldsymbol{c}_j^{(k-1)}), \\ \boldsymbol{\omega}_{\boldsymbol{D}}^{j} = \nabla_{\boldsymbol{d}_j} Q(\boldsymbol{Z}^{(k)}) - \nabla_{\boldsymbol{d}_j} Q(\boldsymbol{C}^{(k)}, \boldsymbol{V}_j^{(k)}) - \lambda_j^k(\boldsymbol{d}_j^{(k)} - \boldsymbol{d}_j^{(k-1)}), \end{cases} \tag{22}$$

*and $(\boldsymbol{T}_j^{(k)}, \boldsymbol{V}_j^{(k)})$ is defined in (20). Then, $\boldsymbol{\omega}^k := (\boldsymbol{\omega}_{\boldsymbol{C}}^{(k)}, \boldsymbol{\omega}_{\boldsymbol{D}}^{(k)}) \in \partial H(\boldsymbol{Z}^{(k)})$ and there exists a constant $\rho > 0$, such that*

$$\|\boldsymbol{\omega}^k\|_F \leq \rho \|\boldsymbol{Z}^{(k)} - \boldsymbol{Z}^{(k-1)}\|_F.$$

**Lemma 3.** *The sequence $\{\boldsymbol{Z}^{(k)}\}_{k \in \mathbb{N}}$ satisfies the Continuity condition (P3).*

For the property (P4), see [9] for the definition. An important class of functions that satisfies the Kurdyka-Lojasiewicz property is the so-called semi-algebraic functions [9].

**Definition 2.** *(Semi-algebraic sets and functions [9, 3]) 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*

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

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

**Theorem 3.** *([9]) Let $f$ is a proper and lower semicontinuous function. If $f$ is semi-algebraic then it satisfies the K-L property at any point of* $\operatorname{dom} f$.

**Lemma 4.** *All the function $F(\boldsymbol{C})$, $Q(\boldsymbol{Z})$ and $G(\boldsymbol{D})$ defined in (11) are semi-algebraic functions. Moreover, $H(\boldsymbol{Z}) = F(\boldsymbol{C}) + Q(\boldsymbol{Z}) + G(\boldsymbol{D})$ is the semi-algebraic function.*

## References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. (2006)

2. A.Rakotomamonjy: Direct optimization of the dictionary learning. IEEE Trans. Signal Process. (2013)
3. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality. Math. Oper. Res. 35(2), 438–457 (2010)
4. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. Math. Program. Ser. A. 137(1-2), 91–129 (2013)
5. Bao, C., Cai, J., Ji, H.: Fast sparsity-based orthogonal dictionary learning for image restoration. In: ICCV (2013)
6. Bao, C., Ji, H., Quan, Y., Shen, Z.: $\ell_0$ norm based dictioanry learning by proximal method with global convergence. In: CVPR (2014)
7. Barchiesi, D., Plumbley, M.D.: Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. IEEE Trans. Signal Process. (2013)
8. Bobin, J., Starck, J.L., Fadili, J.M., Moudden, Y., Donoho, D.L.: Morphological component analysis: An adaptive thresholding strategy. IEEE Trans. Image Process. 16(11) (2007)
9. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. Ser. A. pp. 1–36 (2013)
10. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. (1999)
11. Dai, W., Xu, T., Wang, W.: Dictionary learning and update based on simultaneous code-word optimzation (simco). In: ICASSP. IEEE (2012)
12. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshop of Generative Model Based Vision (WGMBV). IEEE (2004)
13. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. (2001)
14. Jenatton, R., Mairal, J., Bach, F.R., Obozinski, G.R.: Proximal methods for sparse hierarchical dictionary learning. In: ICML (2010)
15. Jiang, Z., Lin, Z., Davis, L.: Learning a dicscriminative dictionary for sparse coding via label consistent K-SVD. In: CVPR (2011)
16. Mailhé, B., Barchiesi, D., Plumbley, M.D.: INK-SVD: Learning incoherent dictionaries for sparse representations. In: ICASSP (2012)
17. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. JMLR (2010)
18. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. In: NIPS (2009)
19. Martínez, A., Benavente, R.: The ar face database. Tech. rep., Computer Vision Center (1998)
20. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends optim. 1(3), 123–231 (2013)
21. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR. IEEE (2010)
22. Rockafellar, R.T., Wets, R.J.B.: Variational analysis: grundlehren der mathematischen wissenschaften, vol. 317. Springer (1998)

23. Schnass, K., Vandergheynst, P.: Dictionary preconditioning for greedy algorithms. IEEE Trans. Signal Process. 56(5), 1994–2002 (2008)
24. Tosic, I., Frossard, P.: Dictionary learning. IEEE Signal Process. Mag. (2011)
25. Tropp, A.: Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory (2004)
26. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse representation for computer vision and pattern recognition. Proc. IEEE 98(6), 1031–1044 (2010)
27. Xu, Y., Yin, W.: A fast patch-dictionary method for the whole image recovery. UCLA CAM report (2013)
28. Zhang, Q., Li, B.: Discriminative K-SVD for dictionary learning in face recognition. In: CVPR (2010)