J. Vis. Commun. Image R. 23 (2012) 182–193

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A novel predual dictionary learning algorithm

Qiegen Liu, Shanshan Wang, Jianhua Luo *

College of Life Science and Technology, Shanghai Jiaotong University, 200240 Shanghai, China

ARTICLE INFO

Article history:
Received 10 September 2010
Accepted 19 September 2011
Available online 25 September 2011

Keywords:
Dictionary learning
Sparse representation
Predual proximal point algorithm
Bregman iteration method
Iterated refinement property
Gradient descent
Majorization-minimization
Image denoising

ABSTRACT

Dictionary learning has been a hot topic fascinating many researchers in recent years. Most of existing methods have a common character that the sequences of learned dictionaries are simpler and simpler regularly by minimizing some cost function. This paper presents a novel predual dictionary learning (PDL) algorithm that updates dictionary via a simple gradient descent method after each inner minimization step of Predual Proximal Point Algorithm (PPPA), which was recently presented by Malgouyres and Zeng (2009) [F. Malgouyres, T. Zeng, A predual proximal point algorithm solving a non negative basis pursuit denoising model, Int. J. Comput. Vision 83 (3) (2009) 294–311]. We prove that the dictionary update strategy of the proposed method is different from the current ones because the learned dictionaries become more and more complex regularly. The experimental results on both synthetic data and real images consistently demonstrate that the proposed approach can efficiently remove the noise while maintaining high image quality and presents advantages over the classical dictionary learning algorithms MOD and K-SVD.

1. Introduction

In the last two decades, more and more studies have focused on dictionary learning, the goal of which is to obtain a set of prototype signals $d_j \in R^M$, henceforth referred to as atoms, which forms a dictionary $D \in R^{M\times J}$ that can be used to represent a particular set of given signals $x_l \in R^M$ by some sparse linear combination of the atoms in the dictionary. Sparse representation of signals under the learned dictionary possesses significant advantages over the prespecified dictionary such as wavelet and Discrete Cosine Transform (DCT) as demonstrated in many literatures [1–3] and it has been widely used in denoising, inpainting, and classification with state-of-the-art results obtained [1–3].

In the sense of mathematics, considering there is a signal x_l in R^M, it can be represented by a linear combination of atoms either exactly as $x_l = D\alpha_l$ or proximately as $x_l \approx D\alpha_l$, where D represents a suitable dictionary and α_l denotes the representation coefficients. Given an input matrix $X = [x_1,\ldots,x_L]$ in $R^{M\times L}$ of L signals, thus the problem can be formulated as an optimization problem over a dictionary $D = [d_1,\ldots,d_J]$ in $R^{M\times J}$ and the sparse representation matrix $\Gamma = [\alpha_1,\ldots,\alpha_L]$ in $R^{J\times L}$, namely

$$\begin{cases} \min\limits_{D,\Gamma} \sum\limits_{l=1}^{L} \|x_l - D\alpha_l\|_2^2 \\ \text{s.t. } \|\alpha_l\|_0 \leqslant K, \quad l = 1,\ldots,L; \quad \|d_j\|_2^2 = 1, \quad j = 1,\ldots,J \end{cases} \tag{1}$$

where K controls the sparseness of the representation and $\|\bullet\|_0$ denotes the l_0 norm which counts the number of nonzero coefficients of the vector. Replacing the l_0 norm with l_1 norm, another two similar objectives could alternatively be formulated as:

$$\begin{cases} \min\limits_{D,\Gamma} \sum\limits_{l=1}^{L} \|\alpha_l\|_1 \\ \text{s.t. } \|x_l - D\alpha_l\|_2 \leqslant \tau, \quad l = 1,\ldots,L; \quad \|d_j\|_2^2 = 1, \quad j = 1,\ldots,J \end{cases} \tag{2}$$

and

$$\begin{cases} \min\limits_{D,\Gamma} \sum\limits_{l=1}^{L} \lambda\|\alpha_l\|_1 + \|x_l - D\alpha_l\|_2^2 \\ \text{s.t. } \|d_j\|_2^2 = 1, \quad j = 1,\ldots,J \end{cases} \tag{3}$$

where τ is the tolerable limit of error in reconstruction, and λ is the regularization parameter tuning the weight between the regularization term and the fidelity term. Though the solutions to Eqs. (1)–(3) need not be the same mathematically, they are similar in essence to what the sparse representation problem aims at achieving. In fact, the solutions of the three problems coincide with each other for appropriate parameter choice of K, τ and λ.

Traditional methods for solving Eqs. (2) and (3) adopt two-step iterative approaches of a sparse coding step where sparse approximations Γ found with D fixed and a dictionary update step where D is optimized based on the current Γ. After initialization of the dictionary D those algorithms keep iterating between the two steps until either they have run for a predefined number of alternating optimizations or a specific approximation error is reached.

* Corresponding author.
E-mail address: jhluo@sjtu.edu.cn (J. Luo).

1047-3203/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2011.09.008

At the sparse coding step, solving Eq. (2) or Eq. (3) with respect to a fixed dictionary D proves to be a (Non-deterministic Polynomial) NP-hard problem [4], one simplest technique to solve which is approximately greedy pursuit algorithm with sub-linear complexity such as the Matching Pursuit (MP) and the successive Orthogonal Matching Pursuit (OMP) [5–7], and another famous pursuit algorithm is the Basis Pursuit (BP) [8,9] and its variants such as FOCal Underdetermined System Solution (FOCUSS) [10,11], LARS-Lasso algorithm [12]. In general, it offers usually better performances in terms of recovery at the price of a computational complexity equivalent to the one of linear programming.

At the dictionary updating step, solving optimization problems Eq. (2) or Eq. (3) over bases D given fixed coefficients Γ reduced to address a least squares problem with quadratic constraints. In general, this constrained optimization problem can be solved by using several methods, including gradient descent followed by re-normalization of each atom such as maximum likelihood (ML) [13,14] and maximum*a posteriori* (MAP) with iterative projection [15], or a dual version derived from its Lagrangian proposed by Lee et al. [8]. Engan et al. for the method of optimal directions (MOD) algorithm [16] have chosen to solve it using the pseudo inverse of Γ. The K-singular value decomposition (K-SVD) of Aharon et al. [1] uses a different strategy where the columns of D are updated sequentially one at a time using a singular value decomposition (SVD) to minimize the approximation error. The updating step is hence generalized K-means since each patch can be represented by multiple atoms and with different weights.

All the traditional methods mentioned above have a common trait that the sequence of dictionaries becomes simpler and simpler regularly. Because the dictionary is updated to adapt to the new residual least square which is produced when sparse coding is done with some least square constraints determined by τ or λ, the dictionary update step amounts to a clustering process which we name as "forward clustering" framework. Very recently, some algorithms turned to the Bayesian techniques for dictionary learning [17,18], in which not only the dictionary D but also some other parameters such as dictionary size J would be inferred by assuming some prior process, however, in a general sense, these approaches can also be attributed to the "forward clustering" framework.

In this paper, we propose a different framework that the dictionary learning process is addressed as a dictionary refinement procedure similar to the idea presented by Osher for total variation (TV) image restoration [19,20]. We improve the sparsity prior firstly by setting the regularization parameter λ bigger or similarly by setting τ smaller, and then add the residual to the least square gradually, so the sequence of dictionaries becomes more and more complex, we call this "inverse refinement" framework.

Our main contributions are listed as follows: we thoroughly analyze the predual proximal point algorithm (PPPA) originally developed by Malgouyres [21], and prove that the algorithm possesses iterated refinement property by a Majorization-Minimization technique, thus PPPA can be integrated into our "inverse refinement" framework. Moreover, we update the dictionary by a simple gradient descent based algorithm after each inner minimization step of PPPA, and compare the proposed predual dictionary learning (PDL) algorithm with MOD and K-SVD for image denoising application.

The paper is organized as follows: in Section 2, we recall and extend PPPA by a dictionary update step, thus PDL algorithm is generated. Then In Section 3, after reviewing some concepts of Bregman iteration method, we prove the iterated refinement property of PPPA and reveal the differences between our proposed method and the traditional method. The PDL results on synthetic data are given in Section 4, and its comparisons with MOD and K-SVD in the application for image denoising are presented in Section 5. Finally, some discussion and conclusions are played in Section 6.

2. The form of the PDL algorithm

In this section, we present the proposed PDL algorithm, which is an extension of PPPA. We first review the details of the original PPPA and then extend PPPA by updating dictionary via a gradient descent after each inner minimization step of it, thus the new PDL algorithm is developed.

2.1. Summary of PPPA

The PPPA is an efficient sparse coding algorithm under fixed dictionary. It is originally developed to solve Eq. (2) with constraining variable α non-negative, $\tilde{D}$ and $\tilde{\alpha}$ set as $\tilde{D} = [D, -D], \tilde{\alpha} = [\max(\alpha^T, 0), \max(-\alpha^T, 0)]^T$. For convenience D and α denote $\tilde{D}, \tilde{\alpha}$ again, and after the subscript variable l is omitted for the sake of clarity, Eq. (2) can then be rewritten as:

$$(\mathrm{D}) \begin{cases} \min\limits_{\alpha \in R_+^{2J}} \alpha^T I_{2J} \\ \text{s.t. } \|x - D\alpha\|_2 \leqslant \tau \end{cases} \tag{4}$$

where I_{2J} is a column vector with unit elements. In Ref. [21], they solved Eq. (4) through a primal–dual framework.

Firstly they proved that the problem (D) of Eq. (4) is indeed equivalent to the dual formation of the optimization problem (P):

$$(\mathrm{P}) \begin{cases} \min\limits_{u} \tau\|u\|_2 - u^T x \\ \text{s.t. } D^T u \leqslant I_{2J} \end{cases} \tag{5}$$

where u and α denote the primal and dual variables respectively.

The corresponding Lagrangian of the problem (P) is

$$L(u, \alpha) \triangleq \tau\|u\|_2 - u^T x + \alpha^T (D^T u - I_{2J}) \tag{6}$$

Then they turned to a Proximal Point Algorithm (PPA) to solve Eq. (5) and computed the following sequence:

$$\begin{aligned} c^{k+1} &= \arg\min_{u \in R^M} L'(u, \alpha, c^k, \beta_k) = \arg\min_{u \in R^M} \beta_k \|u - c^k\|_2^2 + L(u, \alpha) \\ &= \arg\min_{u \in R^M} \beta_k \|u - c^k\|_2^2 + \tau\|u\|_2 - u^T x + \alpha^T (D^T u - I_{2J}) \end{aligned} \tag{7}$$

Notice that now L' respecting to u is strongly convex with modulus $2\beta_k$. In fact, minimizing Eq. (7) for variable u can be computed with closed form formulae:

$$c^{k+1} = \begin{cases} 0, if \|2\beta_k c^k - (D\alpha - x)\|_2 \leqslant \tau \\ \frac{\|2\beta_k c^k - (D\alpha - x)\|_2 - \tau}{2\beta_k \|2\beta_k c^k - (D\alpha - x)\|_2} (2\beta_k c^k - (D\alpha - x)), \text{ otherwise} \end{cases} \tag{8}$$

Finally the remaining difficulty is how to determine α, Malgouyres suggested using a projected gradient based algorithm to update it and looping several steps to approximate the minimization, namely:

$$\begin{aligned} \alpha^{m+1} &= \max\left(0, \alpha^m + \rho \frac{\partial L'}{\partial \alpha}\Big|_{\alpha = \alpha^m}\right) \\ &= \max(0, \alpha^m + \rho(D^T u^m - I_{2J})) \end{aligned} \tag{9}$$

To sum up, the PPPA is a two-level iterative method, the diagram of which is displayed in Table 1. More details can be seen in Ref. [21].

2.2. Dictionary updating

It would be much promising if we update the dictionary D at the same time. If we extend the penalty functional $L'(u, \alpha, c^k, \beta_k)$ in Eq. (7) to be $L'(u, \alpha, c^k, \beta_k, D)$ by introducing a new variable D. Then updating the dictionary will substantially decrease the functional

Table 1
The detail description of the PPPA algorithm.

▲ Outer Loop (loop in k):
  ▲ Inner Loop (loop in m):
    Denote $u^m = 2\beta_k c^k - (D^T\alpha^{k,m} - x)$, then $u^{m+1} = \begin{cases} 0, if \|u^m\|_2 \leqslant \tau \\ \frac{\|u^m\|_2 - \tau}{2\beta_k \|u^m\|_2} u^m, \text{ otherwise} \end{cases}$
    $\alpha^{k,m+1} = \max(0, \alpha^{k,m} + \rho(D^T u^{m+1} - I_P))$
  end while
  $c^{k+1} = u^{m+1}$; $\alpha^{k+1,0} = \alpha^{k,m+1}$
end while

L'. A natural way to update is following the gradient descent direction, so we get the derivative of L' respective to D and it yields:

$$D_k = D_k + \mu \frac{\partial(\sum_l L')}{\partial D}\bigg|_{D=D_k} = D_k + \mu C^{k+1}(\Gamma^{k+1})^T \tag{10}$$

where $C^{k+1} = [u_1^{m+1}, \ldots, u_L^{m+1}], \Gamma^{k+1} = \left[\alpha_1^{k,m+1}, \ldots, \alpha_L^{k,m+1}\right]$.

After the gradient descent, a normalization of dictionary columns is required.

2.3. The diagram of the PDL algorithm

In summary, the proposed PDL algorithm inherits the advantages of PPPA, in addition, it updates dictionary after each inner loop of PPPA. The diagram of PDL is described in Table 2.

3. The property of the PDL algorithm

In this section, we investigate the properties of the proposed algorithm and show the major distinction between our PDL algorithm and the classical dictionary learning algorithms. Firstly, we prove that with the dictionary fixed the PPPA has iterated refinement property, just like Bregman iteration method [19,20], which was not mentioned in Ref. [21]; Secondly, we state that the dictionary sequence generated by our method also has the similar iterated refinement property (i.e. as the iteration proceeds, the dictionary becomes more and more complex). This is an essential difference from other dictionary learning algorithms.

3.1. A review of the Bregman iteration method

Consider a constrained optimization problem of the form:

$$\begin{cases} \min\limits_{\alpha\in\Omega} E(\alpha) \\ \text{s.t. } H(\alpha) = 0 \end{cases} \tag{11}$$

with E and H convex, H differentiable. The so called Bregman iteration method, firstly proposed by Osher [19,20,22,23], is given by:

$$\begin{aligned} \alpha_{k+1} &= \arg\min_{\alpha\in\Omega} B_E^{p_k}(\alpha, \alpha_k) + H(\alpha) \\ &= \arg\min_{\alpha\in\Omega} E(\alpha) - \langle p_k, \alpha - \alpha_k\rangle + H(\alpha) \end{aligned} \tag{12}$$

where $B_E^p(\alpha, z) = E(\alpha) - E(z) - \langle p, \alpha - z\rangle$ denotes Bregman distance between points α and z. $p \in \partial E(z)$ is some subgradient in the subdifferential of E at point z. $\langle\cdot,\cdot\rangle$ denotes the usual duality product.

Concerning the update of p_k from (12), we have $0 \in \partial(B_E^p(\alpha, \alpha_k) + H(\alpha))$, when this sub-differential is evaluated at α_{k+1}, this is

$$0 \in \partial\left(B_E^p(\alpha_{k+1}, \alpha_k) + H(\alpha_{k+1})\right)$$

Since it is assumed that $p_{k+1} \in \partial E(\alpha_{k+1})$ and H is differentiable, p_{k+1} should be chosen as

$$p_{k+1} = p_k - \nabla H(\alpha_{k+1}) \tag{13}$$

where ∇ denotes the derivative operator. In the special case of $H(\alpha) = \frac{1}{4\beta}\|x - D\alpha\|^2$, let $\hat{\alpha}$ be the perfect recovery, we recall several key convergence results from [19,20,22,23].

Proposition 1. *Under the above assumptions, the sequence $\{\alpha_k\}$ obtained from the iteration (12) and (13) satisfies the following two conditions* [20]:

1. $\|x - D\alpha_{k+1}\|^2 \leqslant \|x - D\alpha_k\|^2$
2. $B_E^{p_k}(\hat{\alpha}, \alpha_{k+1}) < B_E^{p_k}(\hat{\alpha}, \alpha_k), if \|x - D\alpha_k\|^2 \geqslant \eta\|x - D\hat{\alpha}\|^2, \forall\eta > 1$

The first one declares that the sequence $\{D\alpha_k\}$ evolves toward$x$; and the second one indicates that $B_E^{p_k}(\hat{\alpha}, \alpha_k)$ strictly decrease monotonically as long as $\|x - D\alpha_k\|^2 \geqslant \eta\|x - D\hat{\alpha}\|^2$. These two properties are very important for various applications such as image denoising ([19,22,23]): Supposing $x = D\hat{\alpha} + n$ is a noisy image, where $D\hat{\alpha}$ is the noise-free image, n is the noise satisfying $\|n\|_2 = \sigma$, then we can construct a stopping rule for the iterative procedure such as the iteration could be stopped at k when $\|x - D\alpha_k\|_2$ satisfies the condition $\|x - D\alpha_k\|_2 < \eta\sigma$ for the first time.

Usually the Bregman iteration method is also called iterated refinement method. In order to describe it more clearly, we let $p_k = D^T s_k$ and reformulate (12) and (13) as follow:

$$\begin{cases} \alpha_{k+1} = \arg\min\limits_{\alpha\in\Omega} E(\alpha) + \frac{1}{4\beta}\|x + s_k - D\alpha\|^2 \\ s_k + x = s_{k+1} + D\alpha_{k+1} \end{cases} \tag{14}$$

As explained in [20,23], the iterative procedure (14) has an intriguing multi-scale interpretation: At each iteration k they add the "small scales" s_k back to x and perform the inner minimization with x replaced by $x + s_k$, which is then decomposed into "large scale" $D\alpha_{k+1}$ and "small scale"s_{k+1}.

Table 2
The detail description of the PDL algorithm.

1: Input: Signal set $X, D_0, \Gamma^0 = [\alpha_1^0, \ldots, \alpha_L^0]$.
2: Output: Dictionary D and sparse matrix $\Gamma = [\alpha_1, \ldots, \alpha_L]$ satisfying Eq. (8)
3: Initialization: Set $D_0, \Gamma^0 = [\alpha_1^0, \ldots, \alpha_L^0]$
4: While satisfying condition 1 (loop in k):
5: While satisfying condition 2 (loop in m)
6: Denote $u^m = 2\beta_k c^k - (D_k\alpha^{k,m} - x)$, then $u^{m+1} = \begin{cases} 0, if \|u^m\|_2 \leqslant \tau \\ \frac{\|u^m\|_2 - \tau}{2\beta_k\|u^m\|_2} u^m, otherwise \end{cases}$
7: Update $\alpha^{k,m+1} = \max\left(0, \alpha^{k,m} + \rho\left(D_k^T u^{m+1} - I_P\right)\right)$
8: end while
9: $c^{k+1} = u^{m+1}, C^{k+1} = [u_1^{m+1}, \ldots, u_L^{m+1}], \alpha^{k+1,0} = \alpha^{k,m+1}, \Gamma^{k+1} = \left[\alpha_1^{k,m+1}, \ldots, \alpha_L^{k,m+1}\right]$
10: Update $D_{k+1} = D_k + \mu C^{k+1}(\Gamma^{k+1})^T$
11: end while

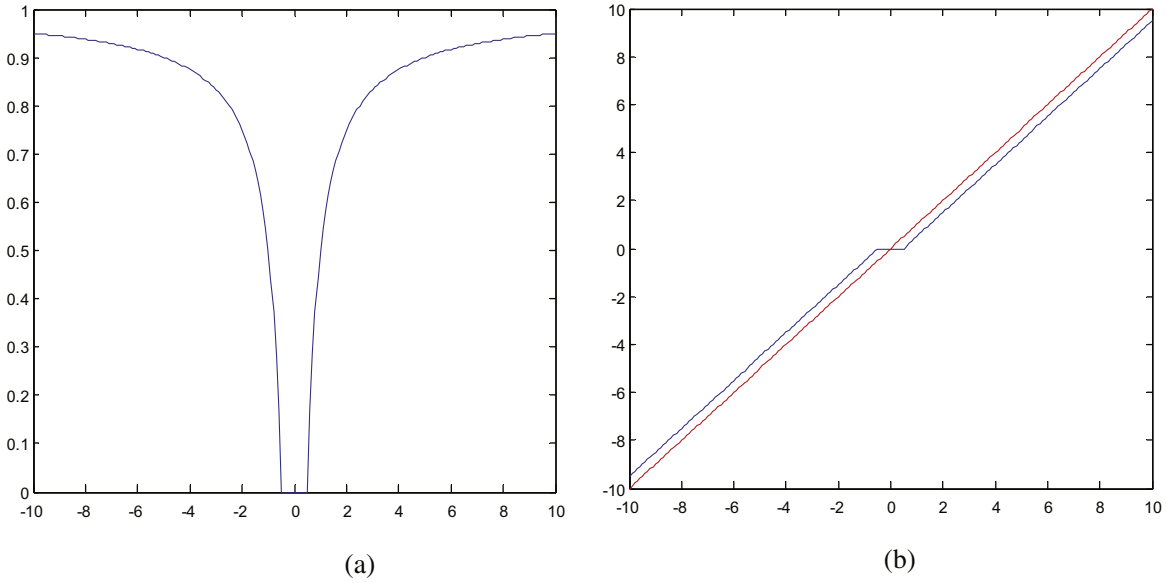

Fig. 1. (a) The thresholding coefficients of the thresholding operator and (b) the thresholding function with variable dimension $M = 1$.

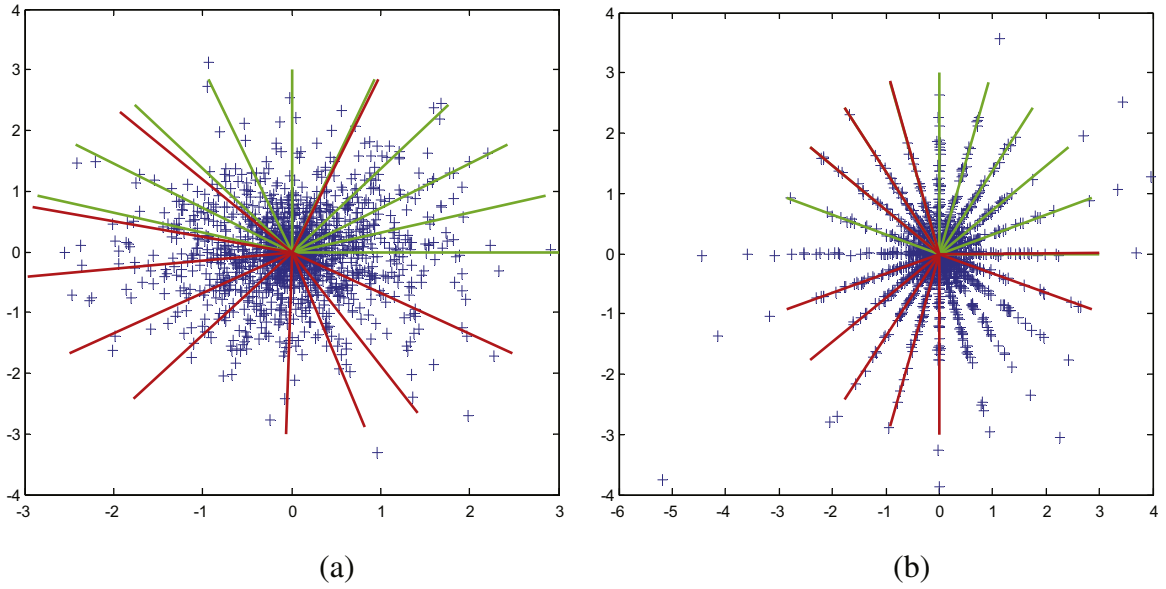

Fig. 2. A display of one run is shown. In the experiment two data samples types are produced, the size of samples is 1500 where each sample is a multiple of one of ten basis vectors plus additive noise, and the samples generated with uniformly distributed i.i.d. coefficients in random and independent locations, the only difference is that one type's coefficients are drawn from Gaussian distribution (a) while the other from Laplacian distribution (b). The green vectors show the true atoms and the red vectors show the learned atoms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. The iterated refinement property of PPPA

For a fixed dictionary, we consider the extreme situation of Eq. (4) by setting $\tau = 0$, then it can be rewritten as follows:

$$\min_{\alpha \geqslant 0} I^T\alpha \text{ s.t. } D\alpha = x \tag{15}$$

Notice that Eq. (15) is exactly the Eq. (11) with $\Omega = \{\alpha \geqslant 0\}, E(\alpha) = I^T\alpha$, and $H(\alpha) = \frac{1}{4\beta}\|x - D\alpha\|^2$, therefore we can use the Bregman iteration method to solve it. Let $p_k = D^T c_k$, then the iteration (12) and (13) become:

$$\begin{cases} \alpha_{k+1} = \arg\min\limits_{\alpha \geqslant 0} \left\{ I^T\alpha + c_k^T(x - D\alpha) + \frac{1}{4\beta}\|x - D\alpha\|^2 \right\} \\ c_{k+1} = c_k + \frac{1}{2\beta}(x - D\alpha_{k+1}) \end{cases} \tag{16}$$

The Bregman iteration formula of Eq. (16) can be solved by using Majorization-Minimization (MM) technique [25,26], which is commonly appeared in optimal numerical community, then we have:

$$\begin{aligned} &\min_{\alpha \geqslant 0} \left\{ I^T\alpha + c_k^T(x - D\alpha) + \frac{1}{4\beta}\|x - D\alpha\|^2 \right\} \\ &= \min_{\alpha \geqslant 0} \left\{ \|x - D\alpha\|^2 + 4\beta I^T\alpha + 4\beta c_k^T(x - D\alpha) \right\} \\ &= \min_{\alpha \geqslant 0} \left\{ \alpha^T D^T D\alpha - 2[D^T x + 2\beta D^T c_k - 2\beta I]^T\alpha + 4\beta c_k^T x \right\} \end{aligned}$$

Defining $M(\alpha) = \alpha^T D^T D\alpha - 2[D^T x + 2\beta D^T c_k - 2\beta I]^T\alpha$ and setting $\alpha_{k,m}$ as point of coincidence, we can find a separable majorizer of $M(\alpha)$ by adding the non-negative function $(\alpha - \alpha_{k,m})^T(\gamma I - D^T D)(\alpha - \alpha_{k,m})$ to $M(\alpha)$, where γ is greater than or equivalent to the maximum

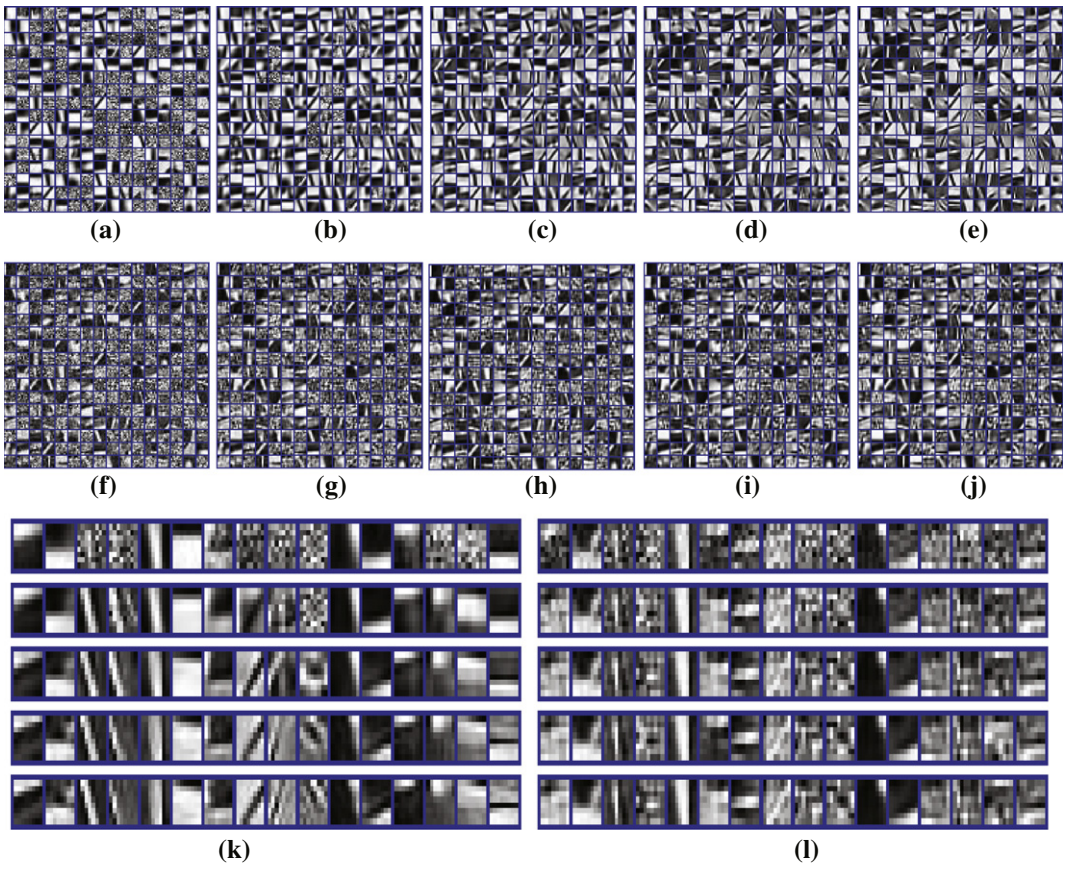

Fig. 3. Two sequences of dictionaries generated by our PDL (Top Line) and K-SVD (Bottom Line) for denoising image 'Boat' in addition with standard deviation $\sigma = 10$. From left to right: the first row (a)(b)(c)(d)(e): the learned dictionaries generated by PDL after 1, 3, 7, 16, 28 iterations; the second row (f)(g)(h)(i)(j): the learned dictionaries generated by K-SVD after 1, 3, 5, 7, 9 iterations. (k) the ninth line of the selected dictionary sequence generated by PDL, (l) the ninth line of the selected dictionary sequence generated by K-SVD (from top to bottom).

eigenvalue of D^TD(the maximum eigenvalue of D is defined by $eig(D) \triangleq \sup_{x\neq 0}\|Dx\|_2/\|x\|_2$), thus a majorizer of $M(\alpha)$ is obtained by

$$Q(\alpha|\alpha_{k,m}) = M(\alpha) + (\alpha - \alpha_{k,m})^T(\gamma I - D^TD)(\alpha - \alpha_{k,m})$$

and we ensure that $Q(\alpha|\alpha_{k,m}) \geqslant M(\alpha)$ and $Q(\alpha_{k,m}|\alpha_{k,m}) = M(\alpha_{k,m})$.

Using the MM approach, the update of α is given by

$$\begin{aligned}\alpha_{k,m+1} &= \arg\min_{\alpha\geqslant 0} Q(\alpha|\alpha_{k,m}) \\ &= \arg\min_{\alpha\geqslant 0}\left\{\gamma\alpha^T\alpha - 2\left[D^Tx + 2\beta D^Tc_k - 2\beta I + \left(\gamma I - D^TD\right)\alpha_{k,m}\right]^T\alpha\right\} \\ &= \arg\min_{\alpha\geqslant 0}\left\{\alpha^T\alpha - 2\left[\alpha_{k,m} + \frac{2\beta}{\gamma}(D^T\left(c_k + \frac{1}{2\beta}(x - D\alpha_{k,m})\right) - I)\right]^T\alpha\right\} \\ &= \arg\min_{\alpha\geqslant 0}\left\{\alpha^T\alpha - 2b_{k,m}^T\alpha\right\}\end{aligned}$$

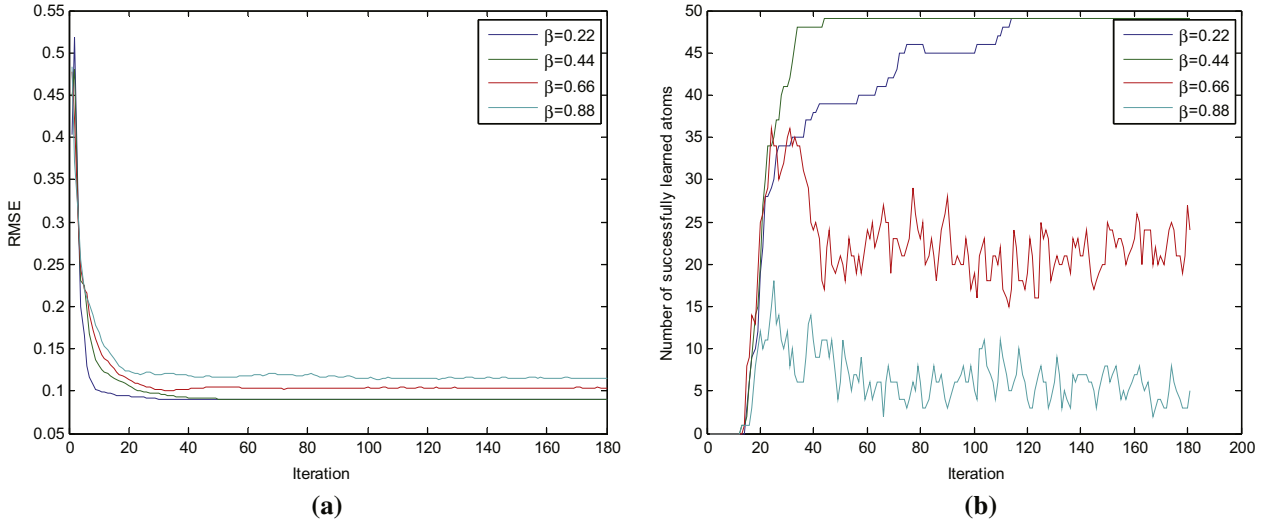

Fig. 4. The evolution of RMSE (a) and successfully detected atom numbers (b) with various β. The target RMSE is 0.1063.

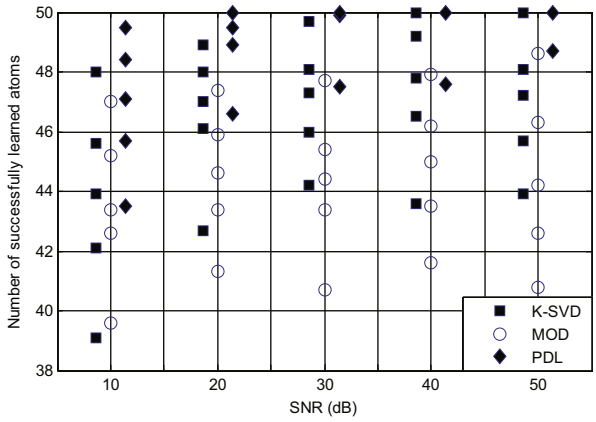

Fig. 5. Comparison results of K-SVD, MOD, and PDL with respect to the reconstruction of the original basis on synthetic signals. For each of the tested algorithms and for each noise level, 50 trials are conducted and their results are sorted. The graph labels represent the mean number of successfully learned basis elements (out of 50) over the ordered tests in groups of ten experiments.

where $b_{k,m} = \alpha_{k,m} + \frac{2\beta}{\gamma}\left[D^T(c_k + \frac{1}{2\beta}(x - D\alpha_{k,m})) - I\right]$

It is easy to see that

$$\alpha_{k,m+1} = \max(0, b_{k,m}) \tag{17}$$

Let $u_{k,m+1} = c_k + (x - D\alpha_{k,m})/2\beta$, then we can reformulate it in two-step alternative way:

$$\begin{cases} u_{k,m+1} = c_k - (D\alpha_{k,m} - x)/2\beta \\ \alpha_{k,m+1} = \max(0, \alpha_{k,m} + \frac{2\beta}{\gamma}(D^T u_{k,m+1} - I)) \end{cases} \tag{18}$$

On the other hand, we rewrite the Inner-Loop in PPPA. i.e. Line 6,7 and 8 in Table 2:

$$\begin{cases} u^m = c^k - (D_k\alpha^{k,m} - x)/2\beta_k \\ u^{m+1} = TH(u^m, \tau/2\beta_k) \\ \alpha^{k,m+1} = \max\left(0, \alpha^{k,m} + \rho\left(D_k^T u^{m+1} - I_P\right)\right) \end{cases} \tag{19}$$

where $TH(u,\mu) = \begin{cases} 0; & |u| \leqslant \mu \\ ((|u| - \mu)/|u|) \cdot u; & |u| > \mu \end{cases}$ is a thresholding operator, the plots of the threshold function and its corresponding threshold coefficient are shown in Fig. 1.

Although Eq. (19) endeavors to solve Eqs. (4) and (18) aims to solve Eq. (15), which is an extreme situation of Eq. (4). By comparing Eq. (18) with Eq. (19) can we still find that Eq. (19) is the same as Eq. (18) except the addition of a thresholding operator. That means Eq. (19) has the same refinement property as Eq. (18) since the thresholding condition doesn't change the property. As mentioned above, we know Eqs. (19) and (18) are the iterative mathematic expression of PPPA and Bregman iteration method respectively. Therefore, it is justified to conclude that PPPA also has the iterated refinement property just as the Bregman iteration method [19,20,22,23]. A comprehensive analysis about Bregman iteration method solving Eq. (4) and its relation with PPPA is supplemented in Appendix.

Remark 1. Interestingly, we state here that the relation between Eqs. (18) and (19) is similar in spirit to the popular soft thresholding addressing with least square problem [27–29]. In particular, the soft thresholding is presented to solve the Eq. (20) by posing the l_1 sparsity priori penalty, or equivalently, it solves Eq. (21)

$$\min_{\alpha} \|x - D\alpha\|_2^2 \tag{20}$$

$$\min_{\alpha} \|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{21}$$

usually, the classical method handling Eq. (20) is the Landweber descent iteration [20,28]:

$$\alpha_{k+1} = \alpha_k + \zeta D^T(x - D\alpha_k) \tag{22}$$

and the soft threshold iteration to solve Eq. (21) is

$$\alpha_{k+1} = ST(\alpha_k + \zeta D^T(x - D\alpha_k), \lambda/2) \tag{23}$$

where $ST(u,\mu) = \begin{cases} 0; & |u| \leqslant \mu \\ (u - sign(u-\mu)\mu) \cdot u; & |u| > \mu \end{cases}$.

3.3. The property of dictionary sequence

Continuing the deduction in the last subsection, we further investigate the learned dictionary updated with our PDL algorithm. Just as it is concluded above that the original PPPA possesses the iterated refinement property, we demonstrate in this part that the property is also reflected in the dictionary updating stage of the proposed PDL algorithm.

As described in subsection 3.1, $D\alpha_{k+1}$ is interpreted as "large scale" and c_{k+1} as "small scale", according to the interpretation, we can find that the update of the current dictionary is in essential to add the prototype of the "small scale" c_{k+1} to the previous dictionary. As the iteration proceeds, the sequence of dictionaries show that the learned dictionary becomes more and more complex.

$$D_{k+1} = D_k + \mu C^{k+1}(\Gamma^{k+1})^T \tag{24}$$

For a comparison, we write the classical update of dictionary as follow:

$$D_{k+1} = D_k + W(X - D_k\Gamma^{k+1})(\Gamma^{k+1})^T \tag{25}$$

when $W = \xi I$, ξ is a constant, Eq. (25) is the ML method [13,14], and when $W = (\Gamma_{k+1}\Gamma_{k+1}^T)^{-1}$, Eq. (25) is the MOD method [16].

Fig. 6. The seven test images used for the various denoising tests ('Boat', 'Barbara', 'Lena', 'Peppers', 'Einstein', 'House', 'Cameraman').

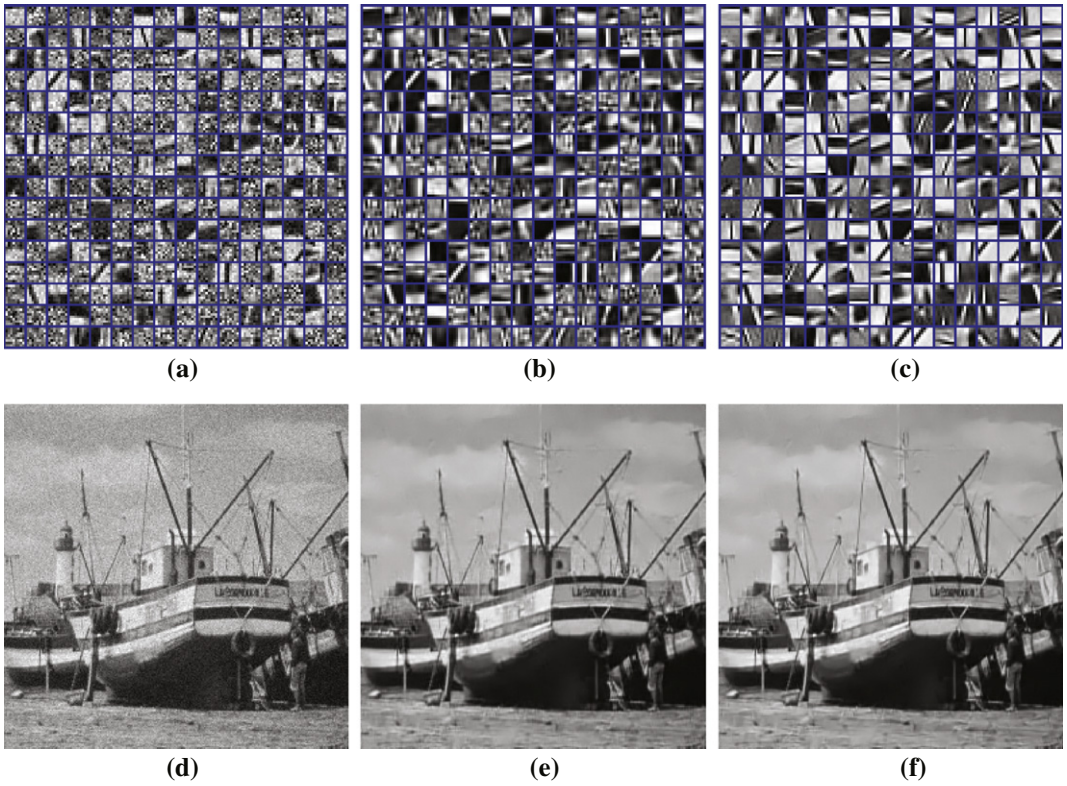

Fig. 7. The denoised result of image "Boat" in addition with standard deviation $\sigma = 10$, Dictionary for 8x8 images patches, Top plots: (a) the initial dictionary, (b) the dictionary trained by K-SVD and (c) the dictionary trained by PDL algorithm. Bottom plots: (d) the reference image, (e) denoised image by K-SVD and (f) denoised image by PDL algorithm.

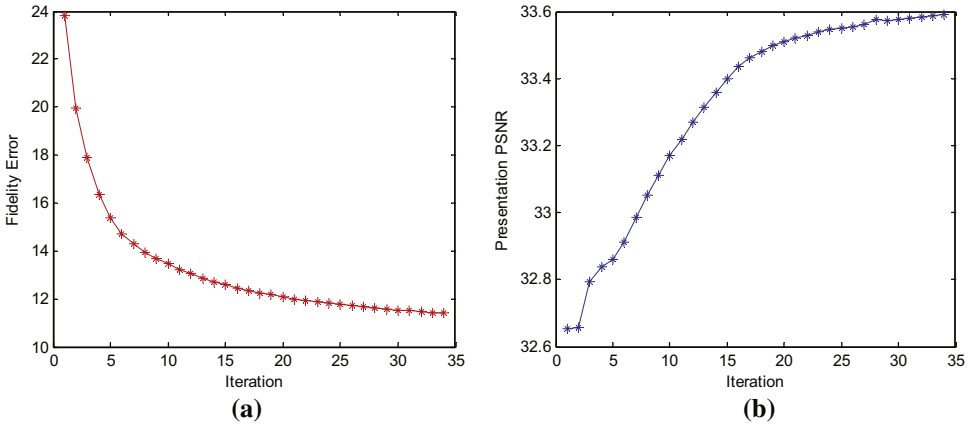

Fig. 8. (a) The Fidelity Error as a function of the iteration and (b) the training representation PSNR versus iterations k respectively.

Eqs. (24) and (25) actually reveal the differences between our method and the classical ones. As the iteration proceeds, the dictionary generated by ours becomes more and more complex; while the dictionary generated by classical ones becomes simpler and simpler. Our iterative procedure is "inverse refinement", which means we enhance the sparsity-prior firstly by setting β a bigger value and then add the residual to the least square gradually; while the classical dictionary learning algorithms whose iterative procedure is "forward", this is to say that they usually support the least square constraints firstly, then pursue the sparsity-prior gradually, so different iterative sparse coding leads to different dictionary updating.

A basic fact must be emphasized is that, as mentioned in many references [1,2], minimizing the functional over coefficient Γ and dictionary D is not convex and many algorithms only converge to a local minimum but not a global one. Therefore, the pathway to minimize the functional will indeed affect the accuracy of the solution. In our current work, we adopt the Bregman iterative strategy, i.e. $D_k \Gamma^k$ converges to X until the constraint in the optimization problem is satisfied. Accordingly, the sparsity of variable $D_k \Gamma^k$ is high at starting point of the iterative procedure and evolves to be lower and lower. Fortunately, this sparsity nature is in favor of dictionary updating. For better interpretability, we illustrate it by a two-dimensional toy example in which the phenomenon is observed similarly as in Ref.

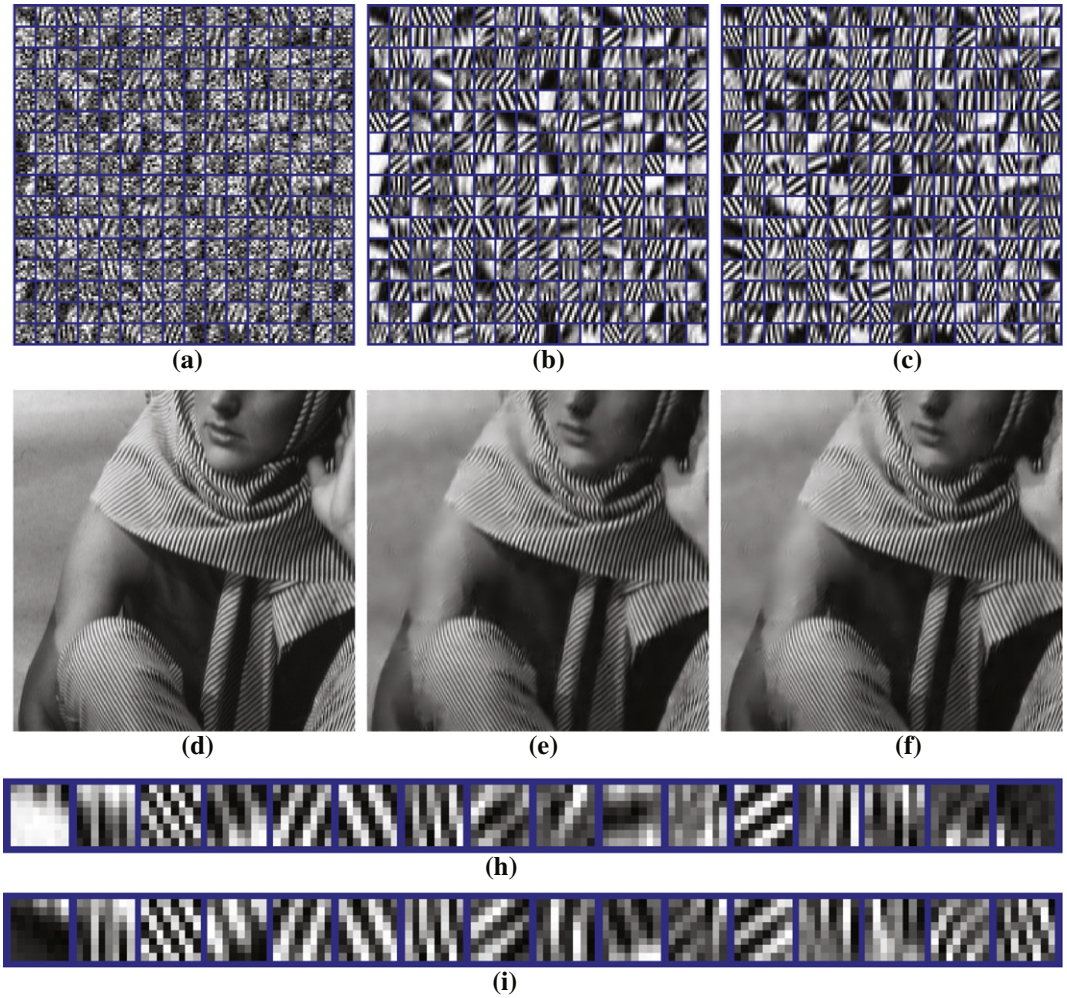

Fig. 9. The denoised result of image "Barbara" in addition with standard deviation $\sigma = 25$, Dictionary for 8x8 images patches, Top plots: (a) the initial dictionary, (b) the dictionary trained by K-SVD and (c) the dictionary trained by PDL algorithm. Middle plots: (d) the reference image, (e) denoised image by K-SVD and (f) denoised image by our PDL algorithm. Bottom plots: (h) the eleventh line of the dictionary (b), (i) the eleventh line of the dictionary (c).

[34]. In the experiment, two data samples types are produced. For both types, the size of samples is 1500 where each sample is a multiple of one of ten basis vectors plus additive noise, and the samples are generated with uniformly distributed i.i.d. coefficients in random and independent locations. The only difference is that one type's coefficients are drawn from Gaussian distribution while the other from a Laplacian distribution. We use MOD over 50 times to measure the identify ability of the algorithm with different realizations of coefficients and noise, and obtain that the average identify ability of the former is 94%, while the latter is almost 100%. Fig. 2 shows a display of one run. This example demonstrates that sparsity indeed enhances the quality of dictionary learning. This explains why our method works better than the traditional approaches.

Fig. 3 depicts two sequences of dictionaries generated respectively by the proposed PDL and K-SVD, which is a representative of the classical dictionary learning algorithms. As can be seen, the dictionary generated by ours is more and more complex; while the dictionary generated by K-SVD is simpler and simpler.

4. Synthetic experiments

As in previously reported works [1,15], we first try the PDL algorithm on artificial data to test its ability for recovering original dictionary that generates the data and then compare it with K-SVD and MOD.

4.1. Test data and criterion of the experiment

We repeat the experiment described in Ref. [1]. A basis $D^{orig} \in R^{20\times 50}$ is generated, consisting of $J = 50$ basis vectors of dimension$M = 20$. Linear combinations $x_1, x_2, \ldots, x_{1500}$ of $k = 3$ basis vectors are computed using uniformly distributed coefficients. We add Gaussian noise to obtain data with varying SNR and the learned basis are produced by applying the K-SVD, MOD and PDL to the data. As in Ref. [1,15] we compare the learned basis to the original basis using the maximum overlap between each original basis vector and the learned basis, i.e., whenever

$$\max_l \left(1 - \left|d_l^{orig} d_l^{learn}\right|\right) \tag{26}$$

is smaller than 0.01, we count this as a success.

4.2. The parameter of the algorithm

We first observe the impact of parameter β on the PDL algorithm, in the case of $SNR = 10$, by setting $\beta = 0.22, 0.44, 0.66, 0.88$ respectively and running the algorithm 180 iterations. With the process of iterations, we investigate the evolution of successfully detected atom numbers and the RMSE that equals to $\frac{1}{\sqrt{ML}}\left\|X - D_k \Gamma^k\right\|_2$. As can be seen in Fig. 4, the RMSE increases with increasedβ, this is consistent with conclusion in Ref. [21]. An interesting phenomenon is

Table 3
The denoising results in db for seven test images and a noise power in the Range [5,100] gray values. for each test setting, three results are provided: our PDL algorithm (TOP), MOD (MIDDLE) AND K-SVD (BOTTOM). The best among each three results is highlighted.

σ_n	'Boat'	'Barbara'	'Lena'	'Peppers'	'Einstein'	'House'	'Cameraman'
5/34.15	**37.79**	**37.88**	**38.07**	**38.08**	**37.31**	**39.44**	**38.02**
	37.40	37.59	37.64	37.58	37.07	39.24	37.61
	37.48	37.63	37.77	37.67	37.14	39.28	37.69
10/28.13	**33.61**	**34.29**	**34.28**	**34.54**	**33.98**	**36.10**	**33.92**
	33.30	34.08	33.90	34.13	33.76	36.07	33.37
	33.34	34.08	33.92	34.16	33.79	36.07	33.41
15/24.61	**31.30**	**32.10**	**32.22**	**32.61**	**32.17**	**34.49**	**31.83**
	31.15	32.98	32.03	32.37	32.02	34.46	31.37
	31.16	31.99	32.04	32.39	32.02	34.46	31.39
20/22.11	**29.75**	**30.61**	**30.72**	**31.12**	**30.81**	**33.31**	**30.15**
	29.68	30.51	30.57	30.93	30.74	33.22	29.94
	29.68	30.52	30.58	30.95	30.75	33.22	29.94
25/20.17	**28.62**	**29.38**	**29.51**	**29.99**	**29.82**	**32.42**	**29.07**
	28.56	29.32	29.38	29.78	29.76	32.36	28.88
	28.58	29.31	29.39	29.79	29.77	32.35	28.91
50/10.61	25.07	24.82	**25.83**	**26.29**	**26.61**	**28.20**	**25.70**
	25.09	24.87	25.80	26.24	26.60	28.11	25.61
	25.11	**24.90**	25.80	26.26	26.59	28.11	25.65
75/10.61	**22.76**	**21.80**	**23.46**	**23.52**	**24.68**	**25.24**	**23.61**
	22.73	21.70	23.45	23.48	24.64	25.19	23.53
	22.74	21.71	23.45	23.49	24.67	25.20	23.58
100/8.13	**21.73**	20.24	**22.22**	**21.65**	22.99	**23.78**	**21.80**
	21.67	20.23	22.14	21.62	**23.00**	23.66	21.71
	21.69	**20.24**	22.14	21.63	22.99	23.66	21.72

that the stability of detected atom numbers decreases with β increased, it seems that the number of successfully detected atoms is gradually increasing when the value of β is very small, so in practical implementation, we can set β relatively smaller when without any priori information of it. We set $\beta = 0.45$ and stop the procedure while satisfying $\tau = \sqrt{M}\sigma_n$ and $k \leqslant 100$ in the experiment. An automatic way to determine β will be investigated further. For fair comparison, the number of learning iterations of K-SVD and MOD is set to 100, which is bigger than in [1].

4.3. Comparison results

The ability of recovering the original dictionary is tested for three methods K-SVD, MOD and PDL, and the comparison results are given in this part. We repeat this experiment 50 times with a varying SNR of 10, 20, 30, 40 dB and as well as 50 dB. As in [21], for each noise level we sort the 50 trials according to the number of successfully learned basis elements and order them in groups of ten experiments. Fig. 5 shows the results of K-SVD, MOD and PDL algorithms. As can be seen, our algorithm outperforms both of them especially when the noise level is low and the number of atoms recovered by PDL is much more than that of both.

5. Numerical experiments of image denoising

In this section, we compare the PDL algorithm with MOD and K-SVD algorithms for application of image denoising, the preliminary results are very encouraging. Seven test images (256×256) shown in Fig. 6 are used for the various denoising tests ('Boat', 'Barbara', 'Lena', 'Peppers', 'Einstein', 'House', 'Cameraman'). The whole process for denoising purpose consists of the following three stages: Firstly, 62001 examples of 8×8 pixel patches, which are overlapped with 7 adjacent pixels on all sides, are obtained by dividing a zero-mean Gaussian noise contaminated image. The initial dictionary whose atom is randomly chosen from the training data; Secondly, for these overlapped patches we apply PDL, MOD and K-SVD to obtain approximate results; Finally, multiple approximate patches at the same pixel location are averaged, and thus the denoised image is obtained, the implementation is the same as in Refs. [1,3].

In the experiment, we set $m = 10$ and $\beta = 100$ for PDL algorithm, and the iteration repeated until the error $\tau = \sqrt{M}\sigma_n$ is satisfied; For MOD and K-SVD algorithms, each patch is extracted and sparse-coded using OMP implementation (error-constrained) [2,3,30], the target error $\tau = C\sqrt{M}\sigma_n$, where $C = 1.15$ is set default, and the iteration number is set to 10. After the learning procedure, the samples are sparse-coded using OMP again under the learned dictionary, this leads to an approximate patch with reduced noise. We use an existing code taken from http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip.

5.1. The learned dictionary

We show the dictionary learned by PDL algorithm and demonstrate its behavior and properties in comparison with K-SVD algorithm, the results are demonstrated in the circumstances of $\sigma = 10$ and $\sigma = 25$. Fig. 7 plots the initial dictionary, the dictionary trained by K-SVD and our PDL algorithm, and the corresponding denoised results of image "Boat" with $\sigma = 10$. As shown, our obtained dictionary is essentially different from that of K-SVD. Comparing with the initial dictionary, each atom of the dictionary learned by K-SVD and PDL seems like a "clustering center", and stands for the pixel patches with very similar properties. While our PDL algorithm recovers more regular prototypes from the original image through the iterated refinement procedure, and the corresponding atoms seem to much "cleaner".

Fig. 8 shows the fidelity error and training representation PSNR[1] values versus iterations$k$ respectively. We see that the fidelity error

[1] Peak Signal-to-Noise ratio (PSNR) as quality measure which is defined in [31] as: $PSNR = 20\log_{10}(255/\|I - \tilde{I}\|_2)$, where I and $\tilde{I}$ are the original and denoised image respectively.

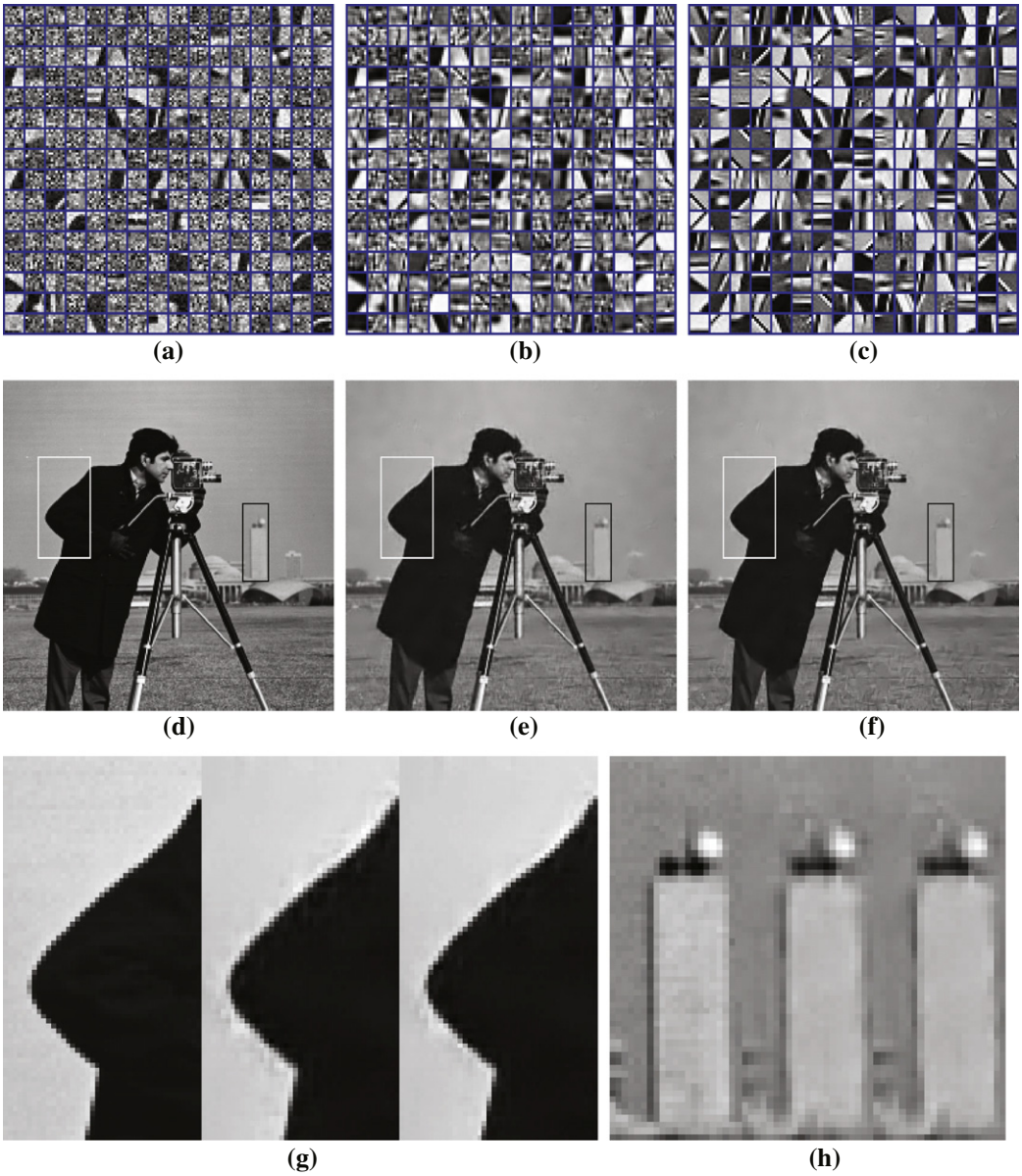

Fig. 10. The denoised result of image "Cameraman" with $\sigma = 15$, Dictionary for 8x8 images patches. Top plots: (a) the initial dictionary, (b) the dictionary trained by K-SVD and (c) the dictionary trained by PDL algorithm. Middle plots: (d) the reference image, (e) denoised image by K-SVD and (f) denoised image by PDL algorithm. Bottom plots: (g) the sub-image contained in the white block of (d), (e) and (f) respectively, (h) the sub-image contained in the black block of (d), (e) and (f) respectively.

decreases with the processes of iteration, this phenomenon confirms our theoretical analysis that PDL contains the refinement property, so we can stop the procedure when it satisfies $\frac{1}{\sqrt{ML}}\|X - D_k\Gamma^k\|_2 = \sigma_n$ at the first time. On the other hand, the PSNR increases with the iterative procedure before arriving at this stopping point, our numerical experiments illustrate that this strategy is very useful. Besides, one can observe that curves of both the Fidelity Error and representation PSNR changes quickly during the first few iterations, this is also verified by the fact that PDL changes the initial dictionary drastically in the first few stages (e.g. $k = 1, 2, 3, \ldots, 13$), which is demonstrated in Fig. 3. In Fig. 3 the learned dictionaries generated at 1, 3, 7, 16, 28 iterations are depicted in the top line. We can observe that from 1 to 7 iterations most of the edge objective atoms are constructed and furthermore from 16 to 28 iterations more and more details are added to the atoms.

Fig. 9 displays the dictionary and the corresponding denoised results of image "Barbara" with $\sigma = 25$. Similar phenomenon is also observed, the dictionary learned by PDL algorithm is "cleaner" than that of K-SVD algorithm. Since there are plenty of repeatedly similar textures in the reference image, the dictionaries in (b) and (c) learned by both methods are very similar, but some differences still can be observed between them. A closer comparison is illustrated in (h) and (i), among which 16 atoms are located in the eleventh line of the dictionary (b) and (c) respectively. Seen from left to right, we find that the last two atoms of (i) learned by PDL are not representable in (h) learned by K-SVD. This indicates that our method has more atom recovery ability, which will be reflected in terms of the denoised results, we will discuss them in next part.

5.2. Results of image denoising

The representation PSNRs in different noise power level for test images using PDL, MOD and K-SVD are listed in Table 3. It can be seen that the PSNRs obtained by K-SVD gain a bit higher than that of MOD generally due to its more complicated and powerful dictionary updating stage. On the other hand, though the dictionary updating stage of PDL is simpler than MOD and K-SVD, the first

Table 4
The average computational load of the K-SVD aND PDL algorithms in different sigma noise level.

σ_n	10	25	100
K-SVD	784.16 s	195.93 s	53.74 s
PDL	448.47 s	316.18 s	78.62 s

sparisty and second refinement strategy makes the proposed method outperform both of them for almost all of noise levels and a variety of image types, especially for image "Lena", "Peppers" and "Cameraman". A striking example is displayed in Fig. 10. Fig. 10 plots the initial dictionary, the dictionary trained by K-SVD and our PDL algorithm, and the corresponding denoised results of image "Cameraman" with $\sigma = 15$. The image denoised by PDL has better visual quality. In particular, the proposed PDL well preserves the image edges and removes the noise without introducing too many artifacts. To facilitate the visual assessment of images quality, in Fig. 10(d)–(f) small regions of the physical image are boxed, in which we clearly observe the differences of the edge as well as the noise and those images contain. We can see that the Fig. 10(g) shows the K-SVD introduces some artifacts and noise while our method does not. What's more, the small boxes of Fig. 10(h) also show the K-SVD makes the edge blurred but the proposed method still keeps most part of the edge.

It should be remarked that if DCT is chosen as the initial dictionary just as in Ref. [2], the three methods will all perform better; meanwhile we can still get the same conclusion that our method is more outstanding than the other two approaches. Nevertheless, in order to test the denoising ability for universal situations, we have chosen a random matrix as initial dictionary in this paper.

5.3. Computational complexity

We return to the computational complexity issue in this part. As analyzed in Section 3, the PPPA we extend in PDL algorithm is Bregman iteration method plus a thresholding operator. Fortunately, the equivalence between Bregman iteration method and the augmented Lagrangian (AL) algorithm was stated in [20], and the AL algorithm has proves to converge to an optimal at a linear rate whose ratio is, roughly speaking, proportional to β_∞ if β_∞ is small (the convergence is super-linear if $\beta_\infty \to 0$) [24]. In addition with the updating of dictionary via a simple gradient descent, we would expect the computational complexity of the whole PDL algorithm to be very promising. Table 4 compares the average computational load of the two algorithms in different noise levels. It can be seen that the computational time of K-SVD is almost two times of ours in circumstance of $\sigma_n = 10$. For K-SVD method, due to its coefficients solved by OMP implementation (error-constrained), so the lower of noise level is, the more computational time it needs. Thanks to the stop criteria we have used, our PDL approach also has such tendency.

6. Discussion and conclusion

In this paper, we propose an "inverse refinement" framework for dictionary learning. In contrast to the traditional "forward clustering" framework, the sequence of dictionaries learned by our framework is iteratively refined and seems to more and more complex regularly. In this framework, we extend PPPA by updating dictionary via a gradient descent after each inner minimization step of it. The results obtained from the numerical experiment are quite competitive with or even superior to the state-of-the-art algorithms. We should emphasize that we use PPPA here only to prove the efficiency of our framework, and the dictionary is updated by gradient descent because of its simplicity. In fact, a variety of techniques can be integrated into our framework for further improvement purposes.

- Since the nature of our framework has the "iterated refinement" property, as proved in [20], the Bregman iterative method is equivalent to the well-known augmented Lagrangian method (i.e., the method of multipliers), so any techniques of the augmented Lagrangian method have the potential to be recast into our framework besides PPPA.
- At the dictionary update stage of the framework, for instance, similar to an online dictionary learning algorithm developed by Mairal [32], we can impose various step size on each atom of the dictionary. In addition, as discussed in [1,15,27,33], the prior that constraining the columns d_j of D to be unit vectors can be utilized to construct the gradient direction at dictionary updating step.

Acknowledgments

This work was partly supported by High Technology Research Development Plan (863 plan) of PR China under 2006AA020805, the NSFC of China under 30670574, Shanghai International Cooperation Grant under 06SR07109, Region Rhône-Alpes of France under the project Mira Recherche 2008, and the joint project of Chinese NSFC (under 30911130364) and French ANR 2009 (under ANR-09-BLAN-0372-01). We would also like to thank the anonymous reviewers for their helpful suggestions in the revision of this manuscript.

Appendix A

In the appendix we prove that Eq. (4) can be solved by Bregman iteration method (or its equivalent formation augmented Lagrangian (AL) [20]). Firstly we transform Eq. (4) into the formulation as following:

$$\min_{\alpha \geqslant 0, z} I^T\alpha + \delta_\tau^2(x - D\alpha) \quad \text{where } \delta_\tau^2(z) = \begin{cases} 0, & \text{if } \|z\|_2 \leqslant \tau \\ +\infty, & \text{otherwise} \end{cases} \tag{27}$$

Similarly as we have done in subsection 3.2, we give:

$$\begin{cases} \{\alpha_{k+1}, z_{k+1}\} = \arg\min_{\alpha \geqslant 0, z} \{I^T\alpha + \delta_\tau^2(z) - c_k^T(D\alpha + z - x) + \frac{1}{4\beta}\|D\alpha + z - x\|_2^2\} \\ c_{k+1} = c_k + \frac{1}{2\beta}(-D\alpha_{k+1} - z_{k+1} + x) \end{cases} \tag{28}$$

Secondly, we consider the iterative scheme of variables z and c in Eq. (28). In this situation, the minimization with respect to z can be computed as following:

$$\begin{aligned} z_{k+1} &= \arg\min_z \left\{ \delta_\tau^2(z) - c_k^T(x - z - D\alpha_{k+1}) + \frac{1}{4\beta}\|x - z - D\alpha_{k+1}\|_2^2 \right\} \\ &= \arg\min_z \left\{ \delta_\tau^2(z) + \frac{1}{4\beta}\|z - (x - D\alpha_{k+1} + 2\beta c_k)\|_2^2 \right\} \end{aligned}$$

Note that if $\|x - D\alpha_{k+1} + 2\beta c_k\|_2 \leqslant \tau$, then $z_{k+1} = x - D\alpha_{k+1} + 2\beta c_k$; while if $\|x - D\alpha_{k+1} + 2\beta c_k\|_2 > \tau$, then the optimal solution can be attained in the boundary curve condition: $\|z_{k+1}\|_2 = \tau$, i.e. $z_{k+1} = \frac{x - D\alpha_{k+1} + 2\beta c_k}{\|x - D\alpha_{k+1} + 2\beta c_k\|_2}\tau$. To sum up, the optimal solution can be expressed explicitly as $z_{k+1} = \begin{cases} \tau\frac{u_k}{\|u_k\|_2}, & \text{if } \|u_k\|_2 \geqslant \tau \\ u_k, & \text{otherwise} \end{cases}$, where $u_k = x - D\alpha_{k+1} + 2\beta c_k$.

At the same time, $c_{k+1} = c_k + \frac{1}{2\beta}(-D\alpha_{k+1} - z_{k+1} + x) = \frac{1}{2\beta}[-z_{k+1} + (-D\alpha_{k+1} + x + 2\beta c_k)] = TH(u_k, \tau/2\beta)$,

where $TH(u,\mu)=\begin{cases}0, & \|u\|_2<\mu\\ \frac{(\|u\|_2-\mu)}{\|u\|_2}u, & \|u\|_2\geqslant\mu\end{cases}$. Finally, we consider the iterative scheme of variable α in Eq. (28).

$$\begin{aligned}\alpha_{k+1}&=\arg\min_{\alpha\geqslant 0}\left\{I^T\alpha+\delta_\tau^2(z_{k+1})-c_k^T(D\alpha+z_{k+1}-x)+\frac{1}{4\beta}\|D\alpha+z_{k+1}-x\|_2^2\right\}\\&=\arg\min_{\alpha\geqslant 0}\left\{I^T\alpha+\frac{1}{4\beta}\|z_{k+1}-(x-D\alpha+2\beta c_k)\|_2^2\right\}\end{aligned}$$

This minimization can be computed by applying Majorization-Minimization (MM) technique similarly as we have done in subsection 3.2. i.e. at each inner loop, we update variable $\alpha_{k,\,m+1}$ as following:

$$\begin{aligned}\alpha_{k,m+1}&=\arg\min_{\alpha\geqslant 0}\left\{I^T\alpha+\frac{1}{4\beta}\|z_m-(x-D\alpha+2\beta c_k)\|_2^2\right.\\&\quad\left.+\left(\alpha-\alpha_{k,m}\right)^T(\gamma I-D^TD)(\alpha-\alpha_{k,m})\right\}\\&=\arg\min_{\alpha\geqslant 0}\left\{\alpha^T\alpha-2\left[\alpha_{k,m}+\frac{2\beta}{\gamma}D^T\left[c_k+\frac{1}{2\beta}(-D\alpha_{k,m}-z_m+x)-\frac{2\beta}{\gamma}I\right]^T\alpha\right\}\right.\end{aligned}\quad(29)$$

If we use the up-to-date value z, i.e. Denoting $u_{m+1}=c_k+\frac{1}{2\beta}(-D\alpha_{k,m}-z_m+x)=TH(u_m,\tau/2\beta)$, then we can rewrite Eq. (29) as following:

$$\begin{aligned}\alpha_{k,m+1}&=\arg\min_{\alpha\geqslant 0}\left\{\alpha^T\alpha-2\left[\alpha_{k,m}+\frac{2\beta}{\gamma}(D^Tu_{m+1}-I)\right]^T\alpha\right\}\\&=\max\left\{0,\alpha_{k,m}+\frac{2\beta}{\gamma}(D^Tu_{m+1}-I)\right\}\end{aligned}$$

In summary, we list the update of all variables in inner loop as following:

$$\begin{cases}u_m=c_k-(D_k\alpha_{k,m}-x)/2\beta\\ u_{m+1}=TH(u_m,\tau/2\beta)\\ \alpha_{k,m+1}=\max\left(0,\alpha_{k,m}+\frac{2\beta}{\gamma}\left(D_k^Tu_{m+1}-I_P\right)\right)\end{cases}$$

This is truly the inner iterative scheme of PPPA algorithm with $\rho=2\beta/\gamma$. So we conclude that PPPA can be viewed as a variant algorithm of solving Eq. (4). It employs MM technique to update variable α of Eq. (28) and at each inner iterative step it use the up-to-date variable z, thus attainting fast convergence.

References

[1] M. Aharon, M. Elad, A.M. Bruckstein, The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representations, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[2] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process 15 (12) (2006) 3736–3745.
[3] M. Aharon, M. Elad, Sparse and redundant modeling of image content using an image-signature-dictionary, SIAM J. Imag. Sci. 1 (2008) 228–247.
[4] D.L. Donoho, M. Elad, V. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, IEEE Trans. Inf. Theory 52 (1) (2006) 6–18.
[5] S. Mallat, Z. Zhang, Matching pursuit in a time-frequency dictionary, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[6] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inf. Theory 50 (10) (2004) 2231–2242.
[7] R. Rubinstein, M. Zibulevsky, M. Elad, Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit, Technical Report – CS Technion, 2008.
[8] H. Lee, A. Battle, R. Rajat, A.Y. Ng, Efficient sparse coding algorithms, in: Adv. NIPS, 2007.
[9] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.
[10] I. Gorodnitsky, B. Rao, Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm, IEEE Trans. Signal Process. 45 (3) (1997) 600–616.
[11] B.D. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection, IEEE Trans. Signal Process. 47 (1) (1999) 187–200.
[12] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2) (2004).
[13] B. Olshausen, D. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1?, Vision Res 37 (23) (1997) 3311–3325.
[14] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature, vol. 381, springer-Verlag, New York, 1996 (6583) pp. 607-609.
[15] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, T. Sejnowski, Dictionary learning algorithms for sparse representation, Neural Comput. 15 (2) (2003) 349–396.
[16] K. Engan, S.O. Aase, J.H. Husoy, Frame based signal compression using method of optimal directions (MOD), in: Proc. ISCS, 1999.
[17] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, L. Carin, Non-parametric bayesian dictionary learning for sparse image representations, Neural Inform. Process. Syst. (NIPS) (2009).
[18] N. Dobigeon, J.Y. Tourneret, Bayesian orthogonal component analysis for sparse representation, IEEE Trans. Signal Process. 58 (5) (2010) 2675–2685.
[19] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration, SIAM J. Multiscale Model. Simul. 4 (2005) 460–489.
[20] W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for l1-minimization with applications to compressed sensing, SIAM J. Imag. Sci. 1 (2008) 142–168.
[21] F. Malgouyres, T. Zeng, A predual proximal point algorithm solving a non negative basis pursuit denoising model, Int. J. Comput. Vision 83 (3) (2009) 294–311.
[22] J. Xu, S. Osher, Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising, IEEE Trans. Image Process. 16 (2) (2007) 534–544.
[23] M. Burger, S. Osher, J. Xu, G. Gilboa, Nonlinear inverse scale space methods for image restoration, Lect. Notes Comput. Sci. 3752 (2005) 25–36.
[24] R.T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Math. Oper. Res. 1 (2) (1976) 97–116.
[25] J. Bect, L. Blanc-F'eraud, G. Aubert, A. Chambolle, A L1-unified variational framework for image restoration. Lecture notes in Computer Science, 2004. Proc. ECCV 2004.
[26] D. Hunter, K. Lange, A tutorial on MM algorithms, The Am. Statist. 58 (2004) 30–37.
[27] M. Elad, Why simple shrinkage is still relevant for redundant representations?, IEEE Trans Inf. Theory 52 (12) (2006) 5559–5569.
[28] I. Daubechies, M. De Friese, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (2004) 3601–3608.
[29] M. Figueiredo, R. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Trans. Image Process. 12 (8) (2003) 906–916.
[30] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Trans. Image Process. 17 (1) (2008) 53–69.
[31] L. Sendur, I.W. Selesnick, Bivariate shrinkage with local variance estimation, IEEE Signal Process. Lett. 9 (12) (2002) 438–441.
[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (1) (2010) 19–60.
[33] Z.S. He, S.L. Xie, L.Q. Zhang, A note on Lewicki–Sejnowski gradient for learning overcomplete representations, Neural Comput. 20 (3) (2008) 636–643.
[34] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Neural Comput. 12 (2000) 337–365.