Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 259863, 11 pages
http://dx.doi.org/10.1155/2013/259863

Research Article

Dictionary Learning Based on Nonnegative Matrix Factorization Using Parallel Coordinate Descent

Zunyi Tang,[1] Shuxue Ding,[2] Zhenni Li,[1] and Linlin Jiang[3]

[1] *Graduate School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan*
[2] *School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan*
[3] *Department for Student Affairs, University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan*

Correspondence should be addressed to Zunyi Tang; tangzunyi@gmail.com

Received 28 February 2013; Accepted 16 May 2013

Academic Editor: Yong Zhang

Sparse representation of signals via an overcomplete dictionary has recently received much attention as it has produced promising results in various applications. Since the nonnegativities of the signals and the dictionary are required in some applications, for example, multispectral data analysis, the conventional dictionary learning methods imposed simply with nonnegativity may become inapplicable. In this paper, we propose a novel method for learning a nonnegative, overcomplete dictionary for such a case. This is accomplished by posing the sparse representation of nonnegative signals as a problem of nonnegative matrix factorization (NMF) with a sparsity constraint. By employing the coordinate descent strategy for optimization and extending it to multivariable case for processing in parallel, we develop a so-called parallel coordinate descent dictionary learning (PCDDL) algorithm, which is structured by iteratively solving the two optimal problems, the learning process of the dictionary and the estimating process of the coefficients for constructing the signals. Numerical experiments demonstrate that the proposed algorithm performs better than the conventional nonnegative K-SVD (NN-KSVD) algorithm and several other algorithms for comparison. What is more, its computational consumption is remarkably lower than that of the compared algorithms.

1. Introduction

Dictionary learning, building a dictionary consisting of atoms or subspaces so that a class of signals can be efficiently and sparsely represented in terms of the atoms, is an important topic in machine learning, neuroscience, signal processing, and so forth. Since in some applications the nonnegativities of the signals and the dictionary are required, for example, multispectral data analysis [1, 2], nonnegative factorization for recognition [3, 4], and some other important problems [5, 6], the so-called nonnegative dictionary learning becomes necessary. In this paper, we mainly focus on this topic.

In the model of sparse representation of signals, a basic assumption is that using an overcomplete dictionary matrix $\mathbf{W} \in \mathbb{R}^{m\times r}$ that contains r atoms of size $m \times 1$ for columns, $\{\mathbf{w}_i\}_{i=1}^{r}$, each column vector of a signal matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$ can be represented as a linear combination of very few, which is meant by the terminology of sparse, atoms $\mathbf{w}_i$ of dictionary $\mathbf{W}$. Here, the term "overcomplete" means $m < r$. $\mathbf{Y} = \mathbf{WH}$ or $\mathbf{Y} \approx \mathbf{WH}$ satisfying $\|\mathbf{Y} - \mathbf{WH}\|_2 \leq \varepsilon$ are two ways to represent $\mathbf{Y}$. The corresponding matrix $\mathbf{H} \in \mathbb{R}^{r\times n}$ that contains the representation coefficients of signals $\mathbf{Y}$ is called the coefficient matrix. For dictionary $\mathbf{W}$, it can be either generated by a prespecified set of functions or learned by a given set of training signals. In practices [7, 8], learning a dictionary has proved to be critical to achieve superior results in the domains of signal and image processing.

Naturally, the problem of finding a dictionary and its sparse representation with the fewest number of atoms can be modeled by using the ℓ^0-norm. Considering the fact that the ℓ^0-norm optimization problem is generally NP-hard, one frequently used heuristic is the ℓ^1-minimization [9]. A series of studies has led to many dictionary learning algorithms. Several classical algorithms include LARS [10], K-SVD [11], ILS-DLA [12], ODL [13], and RLS-DLA [14].

Although these algorithms are very efficient in general, they are not always suitable for learning a nonnegative dictionary from nonnegative signals. For example, a nonnegative variant of K-SVD, which is termed "NN-KSVD" [15], is not as efficient as K-SVD because the negative elements generated in a dictionary matrix are intentionally set to zero to guarantee nonnegativity as the dictionary updates.

In recent years, nonnegative matrix factorization (NMF) [2, 16] has been widely applied to data analyses having nonnegativity constraints since NMF can factorize a nonnegative matrix into a product of two nonnegative factor matrices with different properties. Intuitively, NMF is similar to sparse representation of nonnegative signals to some extent. However, the standard NMF algorithms [17] do not impose any constraints on the two factors, except for nonnegativity, which is not sufficient to lead to a sparse enough representation. In order to obtain a sparser representation, various sparsity constrained NMF algorithms have been proposed. Hoyer et al. [18–20] considered enforcing the sparsity of coefficient matrix using ℓ^1-norm. Hoyer [21] also introduced a measure of sparsity based on the ratio of the ℓ^1-norm of a vector to the ℓ^2-norm. Some algorithms imposed sparsity constraints by using ℓ^2-norm [5, 22, 23]. Peharz et al. [24, 25] presented sparse NMF algorithms that constrain the ℓ^0-(pseudo-) norm of the coefficient matrix. In addition, several approaches based on other types of constraints, such as nonsmoothness constraint [26], squared ℓ^1-norm penalization [27], and mixed-norm [28], have been proposed recently.

Inspired by the sparsity constrained NMF, in this paper we present a new method for learning a nonnegative overcomplete dictionary for sparse representation of nonnegative signals. Differently from the optimization strategies used in the conventional sparsity constrained NMF, this method employs the coordinate descent strategy [29] and extends it to multivariable case for optimizing multiple independent variables in factors, thus resulting in the so-called parallel coordinate descent strategy. We present the update rules based on the new strategy and develop an algorithm, which is termed as the parallel coordinate descent dictionary learning (PCDDL) algorithm, to solve our objective problem. The proposed algorithm is very efficient since the objective problem has been cast as two sequential optimal problems of quadratic functions not involving the complicated calculations inherent to factorization. Through experimental evaluations, we have observed that the proposed algorithm achieves the best rate of atom recovery compared with the conventional algorithms [15, 18, 21, 25]. In addition, its performance is robust even if noise is quite heavy. Furthermore, the computation cost of our algorithm is much lower than that of other algorithms because it does not involve the complicated calculations.

The remaining part of the paper is organized as follows. In Section 2, we formulate the nonnegative dictionary learning problem. In Section 3, we describe the proposed PCDDL algorithm for nonnegative dictionary learning. In Section 4, we report the results of numerical experiments using PCDDL and compare these results with those of several other algorithms. These experiments involve two groups of synthetic datasets and two preliminary applications involving image processing. Finally, in Section 5, we draw our conclusions and discuss related research topics for the future.

2. Problem Formulation

Given a vector $\mathbf{y} \in \mathbb{R}^m$, whose components are a group of signals, we are now concerned with its sparse representation over an overcomplete dictionary $\mathbf{W} \in \mathbb{R}^{m\times r}$, each column of which is referred to an atom. That is, we attempt to find a linear combination of only few atoms, which can be close to $\mathbf{y}$ in value. To avoid trivial solutions, $\mathbf{W}$ is restricted to the set $\mathbb{C}$, which is defined as

$$\mathbb{C} \triangleq \left\{\mathbf{W} = \left[\mathbf{w}_1, \dots, \mathbf{w}_n\right] : \mathbf{w}_j^T \mathbf{w}_j = 1, \forall j = 1, \dots, n\right\}. \quad (1)$$

For a training set of n signals $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$, dictionary learning can be formulated as the following optimization problem:

$$\min_{\mathbf{W}\in\mathbb{C},\mathbf{H}} \frac{1}{n}\sum_{i=1}^{n} \mathscr{F}_i\left(\mathbf{h}_i, \mathbf{W}\right), \quad (2)$$

where $\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_n\}$ and

$$\mathscr{F}_i\left(\mathbf{h}_i, \mathbf{W}\right) = \frac{1}{2}\left\|\mathbf{y}_i - \mathbf{W}\mathbf{h}_i\right\|_2^2 + P\left(\mathbf{h}_i, \lambda\right). \quad (3)$$

Here $P(\mathbf{h}_i, \lambda)$ is a penalty function with $\lambda > 0$, which is a tuning parameter controlling the tradeoff between the approximation error $(1/2)\|\mathbf{y}_i - \mathbf{W}\mathbf{h}_i\|_2^2$ and the penalty function $P(\mathbf{h}_i, \lambda)$.

Naturally, the problem of learning a dictionary $\mathbf{W}$ and finding a sparse representation $\mathbf{h}_i$ can be modeled by using the ℓ^0-norm, defining $P(\mathbf{h}_i, \lambda)$ as the ℓ^0-norm of $\mathbf{h}_i$; namely, $P(\mathbf{h}_i, \lambda) = \lambda\|\mathbf{h}_i\|_0$. However, the resulting optimization problem is usually NP-hard. Considering this difficulty, one frequently used heuristic is the ℓ^1-norm; that is, $P(\mathbf{h}_i, \lambda) = \lambda\|\mathbf{h}_i\|_1$ [9].

With the use of the ℓ_1-norm, the dictionary learning problem is expressed as follows:

$$\min_{\mathbf{W}\in\mathbb{C},\mathbf{H}} \frac{1}{n}\sum_{i=1}^{n} \left\{\frac{1}{2}\left\|\mathbf{y}_i - \mathbf{W}\mathbf{h}_i\right\|_2^2 + \lambda\left\|\mathbf{h}_i\right\|_1\right\}. \quad (4)$$

Noted that it is allowed to take different values of λ for different penalty functions $P(\mathbf{h}_i, \lambda)$. For the sake of simplicity, however, we assume here that the same λ is applied to every penalty function. Thus, (4) can be also rewritten as a matrix factorization problem with a sparsity penalty,

$$\min_{\mathbf{W}\in\mathbb{C},\mathbf{H}} \frac{1}{2}\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_F^2 + \lambda\|\mathbf{H}\|_{1,1}, \quad (5)$$

where $\|\mathbf{H}\|_{1,1}$ denotes the ℓ_1-norm of the matrix $\mathbf{H}$, that is, the sum of the ℓ^1-norm of each column vector of the matrix $\mathbf{H}$.

Furthermore, if $\mathbf{Y}$ is nonnegative and factors $\mathbf{W}$ and $\mathbf{H}$ are both limited to be nonnegative, then the process is called nonnegative dictionary learning, which can be formulated as,

$$\begin{aligned}\min_{\mathbf{W}\in\mathbb{C},\mathbf{H}} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_F^2 + \lambda\|\mathbf{H}\|_{1,1} \\ \text{subject to} \quad & \mathbf{W} \geq 0, \ \mathbf{H} \geq 0.\end{aligned} \quad (6)$$

To solve the problem in (6), a natural strategy is to optimize between $\mathbf{W}$ and $\mathbf{H}$ alternatively. That is, minimize one while keeping the other fixed. The NN-KSVD algorithm [18] and some NMF algorithms including NNSC, NMFSC, and NMFℓ^0-H, just solve the problem in such a way.

3. The Proposed Method

3.1. Parallel Coordinate Descent Dictionary Learning (PCDDL). To solve the objective problem (6), we first employ alternating update strategy, that is, updating one of two factors while fixing the other. In the optimization of each factor, we propose optimizing each component in the factor one by one by generalizing the coordinate descent strategy [29], rather than optimizing the whole factor at a time as in the standard NMF algorithms [17]. Furthermore, we found that (6) can be separated into column-wise or row-wise subproblems, and each subproblem can just be solved alternately and explicitly by utilizing the properties of solving extreme value problem of a quadratic function, so that the whole problem can be solved efficiently.

We here derive the update rules for $\mathbf{H}$ and $\mathbf{W}$ of (6). In terms of the definition and properties of the Frobenius norm, for a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\|\mathbf{A}\|_F^2 = \operatorname{Tr}(\mathbf{A}\mathbf{A}^T) = \operatorname{Tr}(\mathbf{A}^T\mathbf{A})$. $\operatorname{Tr}(\cdot)$ denotes the trace of a square matrix. Thus, the objective function (6) can be decomposed as follows:

$$J = \frac{1}{2}\sum_{j=1}^{n}\mathbf{Y}_{j:}^T\mathbf{Y}_{:j} - \sum_{j=1}^{n}\left[\mathbf{Y}^T\mathbf{W}\right]_{j:}\mathbf{H}_{:j} + \frac{1}{2}\sum_{j=1}^{n}\mathbf{H}_{j:}^T\mathbf{W}^T\mathbf{W}\mathbf{H}_{:j} + \lambda\sum_{i=1}^{r}\sum_{j=1}^{n}\left|\mathbf{H}_{ij}\right|, \tag{7}$$

where $[\mathbf{Y}^T\mathbf{W}]_{j:}$ denotes the jth row of the multiplication of matrices $\mathbf{Y}^T$ and $\mathbf{W}$. Since the elements of $\mathbf{H}$ have nonnegativity, the absolute value operation in (7) can be omitted. If we fix $\mathbf{W}$ in (7), then (7) is a multivariable objective function of $\mathbf{H}_{ij}$ $(i = 1,\ldots,r;\ j = 1,\ldots,n)$. First, let us explain the coordinate descent strategy for a single variable. For (7), we consider optimizing only a single variable $\mathbf{H}_{kj}$, while fixing the other components in $\mathbf{H}$. Thus, we obtain a quadratic function with regard to $\mathbf{H}_{kj}$ as follows:

$$J_{\mathbf{H}_{kj}} = \frac{1}{2}\left[\mathbf{W}^T\mathbf{W}\right]_{kk}\mathbf{H}_{kj}^2 + \mathbf{H}_{kj}\left(\sum_{l=1,l\neq k}^{r}\left[\mathbf{W}^T\mathbf{W}\right]_{kl}\mathbf{H}_{lj} - \left[\mathbf{W}^T\mathbf{Y}\right]_{kj} + \lambda\right), \tag{8}$$

where $[\mathbf{W}^T\mathbf{W}]_{kk}$ denotes the entry in the kth row and the kth column of the multiplication of matrices $\mathbf{W}^T$ and $\mathbf{W}$. $[\mathbf{W}^T\mathbf{W}]_{kk}$ is always positive because it is a diagonal element of Gram matrix $\mathbf{W}^T\mathbf{W}$ (no zero vectors exist in $\mathbf{W}$ here, also). Thus, when $\mathbf{H}_{kj} = ([\mathbf{W}^T\mathbf{Y}]_{kj} - \sum_{l=1,l\neq k}^{r}[\mathbf{W}^T\mathbf{W}]_{kl}\mathbf{H}_{lj} - \lambda)/[\mathbf{W}^T\mathbf{W}]_{kk}$, $J_{\mathbf{H}_{kj}}$ reaches the minimum. Considering the nonnegativity of factor $\mathbf{H}$, $\mathbf{H}_{kj}$ is set to 0 when it is negative. Note that, when updating $\mathbf{H}_{kj}$, the process involves only the elements $\mathbf{H}_{lj,l\neq k}$ of the jth column in $\mathbf{H}$. That is, the optimal value for a given entry of $\mathbf{H}$ does not depend on the other components of the same row containing the entry. Hence, one can optimize all elements of one row in $\mathbf{H}$ at the same time. This can be viewed as optimizing the elements in parallel, that is, parallel coordinate descent strategy for multiple variables. Thus, the update rule for $\mathbf{H}$ of (7) is given as follows:

$$\begin{aligned}\mathbf{H}_{k:}^{*} &= \underset{\mathbf{H}_{k:}\geq 0}{\arg\min}\left\|\mathbf{Y} - \mathbf{W}\mathbf{H}\right\|_F^2 \\ &= \max\left(0, \frac{\mathbf{W}_{k:}^T\mathbf{Y} - \sum_{l=1,l\neq k}^{r}\mathbf{W}_{k:}^T\mathbf{W}_{:l}\mathbf{H}_{l:} - \lambda}{\mathbf{W}_{k:}^T\mathbf{W}_{:k}}\right) \\ &= \max\left(0, \frac{\mathbf{W}_{k:}^T\mathbf{R}_k - \lambda}{\left\|\mathbf{W}_{:k}\right\|_2^2}\right),\end{aligned} \tag{9}$$

where $\mathbf{R}_k = \mathbf{Y} - \sum_{l=1,l\neq k}^{r}\mathbf{W}_{:l}\mathbf{H}_{l:}$.

Similar to the derivation of the update rule for $\mathbf{H}$, one can also obtain the corresponding update rule for $\mathbf{W}$. If fixing $\mathbf{H}$ in (7), then (7) is a multivariable objective function of $\mathbf{W}_{ij}$ $(i = 1,\ldots,m;\ j = 1,\ldots,r)$. For (7), we now consider optimizing only one variable $\mathbf{W}_{ik}$, while fixing the other components in $\mathbf{W}$. We first select the items related to $\mathbf{W}_{ik}$ from (7) and obtain a quadratic function with regard to $\mathbf{W}_{ik}$ as follows:

$$J_{\mathbf{W}_{ik}} = \frac{1}{2}\left[\mathbf{H}\mathbf{H}^T\right]_{kk}\mathbf{W}_{ik}^2 + \mathbf{W}_{ik}\left(\sum_{l=1,l\neq k}^{r}\mathbf{W}_{il}\left[\mathbf{H}\mathbf{H}^T\right]_{lk} - \left[\mathbf{Y}\mathbf{H}^T\right]_{ik}\right). \tag{10}$$

One can find that (10) is very similar to (8). In terms of the properties of a single variable quadratic problem, $J_{\mathbf{W}_{ik}}$ obtains the minimum when $\mathbf{W}_{ik} = ([\mathbf{Y}\mathbf{H}^T]_{ik} - \sum_{l=1,l\neq k}^{r}\mathbf{W}_{il}[\mathbf{H}\mathbf{H}^T]_{lk})/[\mathbf{H}\mathbf{H}^T]_{kk}$. Considering the nonnegativity of factor $\mathbf{W}$, $\mathbf{W}_{ik}$ is set to 0 when it is negative. Similar to the update rule for $\mathbf{H}$, $\mathbf{W}$ in (7) can update by column. Thus, the update rule for $\mathbf{W}$ of (7) is expressed as follows:

$$\begin{aligned}\mathbf{W}_{:k}^{*} &= \underset{\mathbf{W}_{:k}\geq 0}{\operatorname{argmin}}\left\|\mathbf{Y} - \mathbf{W}\mathbf{H}\right\|_F^2 \\ &= \max\left(0, \frac{\mathbf{Y}\mathbf{H}_{:k}^T - \sum_{l=1,l\neq k}^{r}\mathbf{W}_{:l}\mathbf{H}_{l:}\mathbf{H}_{:k}^T}{\mathbf{H}_{k:}\mathbf{H}_{:k}^T}\right) \\ &= \max\left(0, \frac{\mathbf{R}_k\mathbf{H}_{:k}^T}{\left\|\mathbf{H}_{k:}\right\|_2^2}\right).\end{aligned} \tag{11}$$

In addition, for preventing dictionary $\mathbf{W}$ from having arbitrarily large values, each column of $\mathbf{W}$ is normalized to the unit ℓ^2-norm when dictionary $\mathbf{W}$ is updating. Note that the way of maintaining the nonnegativity of two factor matrices in PCDDL is obviously different from that of NN-KSVD. The former can guarantee that the obtained nonnegative solutions are the optimal relative to each column-wise or row-wise updating, but the latter cannot.

```
Require: Data Matrix $\mathbf{Y} \in \mathbb{R}_{+}^{m\times n}$, initial matrices $\mathbf{W} \in \mathbb{R}_{+}^{m\times r}$, $\mathbf{H} \in \mathbb{R}_{+}^{r\times n}$, and $\lambda$;
while stopping criterion not satisfied do
    Computing $\mathbf{P} = \mathbf{Y}\mathbf{H}^{T}$ and $\mathbf{Q} = \mathbf{H}\mathbf{H}^{T}$;
    for $k = 1$ to $r$ do
        $\mathbf{W}_{:k} \leftarrow \max\left(0, \frac{\mathbf{P}_{:k} - \sum_{l=1,l\neq k}^{r} \mathbf{W}_{:l}\mathbf{Q}_{lk}}{\mathbf{Q}_{kk}}\right)$
        Normalizing $\mathbf{W}_{:k} \leftarrow \frac{\mathbf{W}_{:k}}{\|\mathbf{W}_{:k}\|_{2}}$
    end for
    Computing $\mathbf{U} = \mathbf{W}^{T}\mathbf{Y}$ and $\mathbf{V} = \mathbf{W}^{T}\mathbf{W}$;
    for $k = 1$ to $r$ do
        $\mathbf{H}_{k:} \leftarrow \max\left(0, \frac{\mathbf{U}_{k:} - \sum_{l=1,l\neq k}^{r} \mathbf{V}_{kl}\mathbf{H}_{l:} - \lambda}{\mathbf{V}_{kk}}\right)$
    end for
    Using the fixed $\lambda$ or adaptively tuning $\lambda$ according to the change of the sparsity of $\mathbf{H}$;
end while
```

Algorithm 1: PCDDL.

Remark 1. According to the above derivation, it can be observed that our objective function (7) can be cast as two sequential optimal problems of quadratic functions, each of which can be alternately optimized in parallel by the generalized coordinate descent strategy.

Remark 2. The sparsity of $\mathbf{H}$ can be flexibly controlled by tuning the regularization parameter λ.

Remark 3. The method is suitable not only for the case of overdetermined dictionary matrices ($m > r$) but also for the case of underdetermined dictionary matrices ($m < r$), even though these matrices have different physical meanings in different applications.

3.2. Choice of Parameter λ and Summary of Algorithm. In the step of updating $\mathbf{H}$ with a fixed $\mathbf{W}$, the parameter $\lambda > 0$ can be adjusted for controlling the tradeoff between the approximation error $(1/2)\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_{F}^{2}$ and the sparsity of coefficient matrix $\mathbf{H}$ and plays an important role in the proposed algorithm. To steer the solution toward a global, optimal solution, the parameter λ can be determined by two kinds of ways, off-line calibrating and adaptive tuning.

For the first way, one can repeat an experiment with different λ and determine what value for λ is the optimal according to the output results.

For the second way, we give an easy-to-use rule as follows. First, λ should be less than $\|\mathbf{W}_{k:}^{T}\mathbf{R}_{k}\|_{\infty}$ in terms of (9); otherwise $\mathbf{H}_{k:}$ will become a zero vector. We may initialize λ with a very small value, for example, 0.001, which can generally satisfy the above condition. Next, we alternately update $\mathbf{H}$ and $\mathbf{W}$ in terms of (9) and (11) and adjust λ according to the rule defined as follows:

$$\lambda^{(k+1)} = \begin{cases} \lambda^{(k)} + 0.001 & \text{if } S\left(\mathbf{H}^{(k-1)}\right) - S\left(\mathbf{H}^{(k)}\right) < 10^{-3}, \\ & S\left(\mathbf{H}^{(k)}\right) - S^{*} > 10^{-3} \\ \lambda^{(k)} & \text{otherwise,} \end{cases} \tag{12}$$

where $S(\mathbf{H})$ is a sparsity measure, defined as $\|\mathbf{H}\|_{0}/(r \times n)$, which calculates the ratio of the number of nonzero elements and the number of all elements in $\mathbf{H}$. $\lambda^{(k)}$ and $S(\mathbf{H}^{(k)})$ denote the value of λ and the sparsity of $\mathbf{H}$ in the kth iteration, respectively. S^{*} denotes the expected or *a prior* sparsity of $\mathbf{H}$. The rule means that if the sparsity of $\mathbf{H}$ varies very slowly and is far from the expected one, one may appropriately increase the stepsize of λ; otherwise, keep the current λ. Experiments show that the values of λ obtained by the two ways are very close. If λ is self-tuned for adapting to signal, however, more iterations are usually needed for convergence.

According to the analysis above, the proposed PCDDL algorithm for nonnegative dictionary learning is summarized in Algorithm 1.

3.3. Convergence Analysis of PCDDL Algorithm. The standard NMF algorithms [17] belong to two-block convex optimization scheme since each factor can be viewed as a block, and optimizing one of two factors while fixing the other is separately convex. Grippo and Sciandrone analyzed the convergence of the two-block convex optimization problems in [30]. They demonstrated that under the condition of continuously differentiable objective function, a two-block convex optimization algorithm does not require each subproblem to have a unique solution for convergence, and any limit point of the sequence of optimal solutions of two-block subproblems is a stationary point. Obviously, PCDDL is such a two-block convex optimization algorithm, so that we can make analysis of its convergence by using the facts in [30]. During iterations, PCDDL can obtain a sequence of the limit points that can guarantee the reduction of objective function. Additionally, in terms of the definition of ℓ^{1}-norm, the penalty term $\|\mathbf{H}\|_{1,1}$ in (6) can be decomposed into $\sum_{i=1}^{r}\sum_{j=1}^{n}\mathbf{H}_{ij}$ since $\mathbf{H} \geq 0$. Thus, under the conditions of $\lambda > 0$, the objective function (6) is differentiable with respect to $\mathbf{W}$ and $\mathbf{H}$, respectively. The existence of limit points and the differentiability of the objective function in (6) imply that the assumptions of Grippo and Sciandrone's Corollary

[30] are satisfied, so that we can establish that the two-block minimization processes of PCDDL converge.

4. Numerical Experiments

In this section, first we present the results of two experiments using PCDDL with synthetic signals. The aims of these experiments are (1) to test whether the PCDDL algorithm can recover the true dictionary, which is used to generate the test data; and (2) to compare the results with those of other algorithms, such as NNSC (online available: http://www.cs.helsinki.fi/u/phoyer/) [18], NN-KSVD (online available: http://www.cs.technion.ac.il/~elad/) [15], NMFSC (online available: http://www.cs.helsinki.fi/u/phoyer/) [21], and NMFℓ^0-H (online available: http://www.spsc.tugraz.at/tools/nmf-l0-sparseness-constraints) [25]. Next, we apply PCDDL to a conventional digital image processing problem, image denoising, to verify the applicability of the proposed algorithm in a real-world environment. Finally, we carry out an experiment of learning a global-based representation on a face dataset in order to demonstrate the practicality of the proposed algorithm for further large-scale data analysis. In the experiments, all programs were coded in Matlab and were run within Matlab 7.8 (R2009a) on a PC with a 3.2 GHz Intel Core i5 CPU and 4 G of memory.

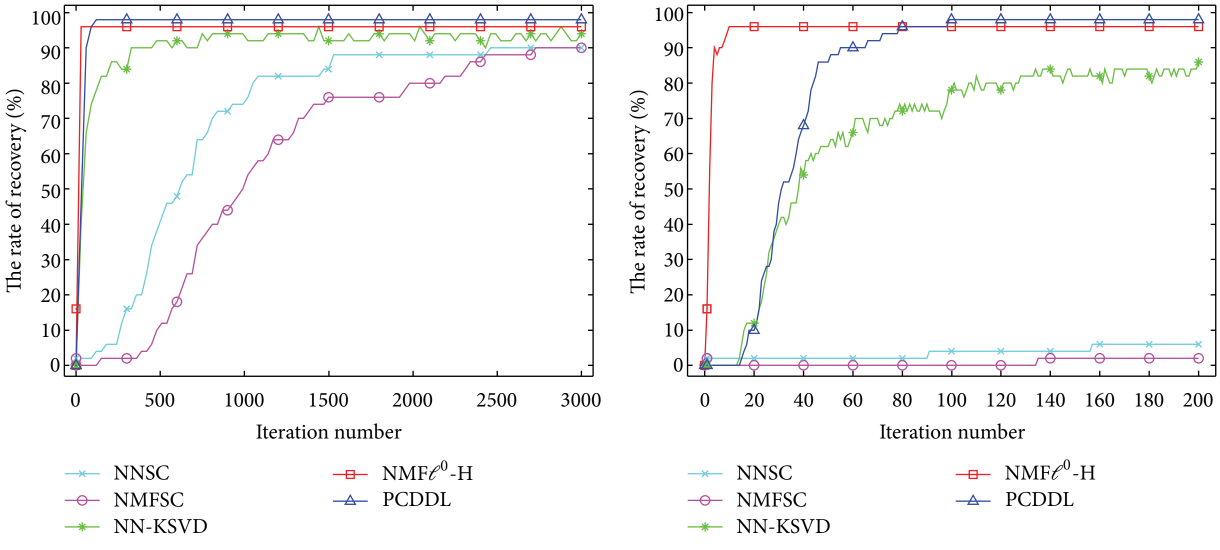
4.1. Recovery Experiment of Random Dictionary. To evaluate the learning capacity of the proposed algorithm for a nonnegative dictionary, we conducted an experiment of recovering a random dictionary from synthetic observation signals generated from the random dictionary. By comparing the recovery rate of the dictionary, adaptability, runtime, and so forth, we assess the algorithms under consideration (see above). The processes are as follows. We generated a stochastic nonnegative matrix of size 20 × 50 with i.i.d. uniformly distributed entries, as described in [11]. Each vector was normalized to unit ℓ^2-norm. The stochastic nonnegative matrix was referred to as the true dictionary $\mathbf{W}$, which was not used in the learning but was used only for evaluation. We then synthesized 1500 test signals $\mathbf{Y}$ of dimension 20, each of which was produced by a linear combination of three different atoms in the true dictionary, with three corresponding coefficients in random and independent positions. We executed NNSC, NMFSC, NN-KSVD, NMFℓ^0-H, and PCDDL on the test signals. For the five algorithms, the initialized dictionary matrices of size 20×50 were composed of the randomly selected parts of the test signals. For NNSC, NMFSC, and PCDDL, the corresponding coefficient matrices were initialized with i.i.d. uniformly distributed random nonnegative entries. NN-KSVD and NMFℓ^0-H do not require a specified coefficient matrix, as they can generate the corresponding coefficient matrix by sparse coding.

Next, we compared the learned dictionaries with the true dictionary. These comparisons were done by sweeping through the columns of the true and the learned dictionaries and finding the closest column (in ℓ^2-norm distance) between the two dictionaries. A distance of less than 0.01 was considered a success. The experiment is similar to the one conducted in [11], except for the nonnegative condition. Obviously, the five iterative algorithms described above have different convergence properties. To provide fair limits on the number of the respective iterations, we executed these algorithms with the same iterations as many times as possible and determined respective iteration number in terms of the results shown in Figures 1, 2, and 3. NNSC and NMFSC, respectively, took about 3000 iterations to reach convergence, while NMFℓ^0-H took only dozens of iterations. In addition, we also considered the runtime of each algorithm as showed in Figure 2. Thus, we set the maximum numbers of iterations for NNSC, NMFSC, NN-KSVD, NMFℓ^0-H, and PCDDL to 3000, 3000, 300, 30, and 500, respectively. Certainly, the iteration of any algorithm can be terminated in advance if it has learned 100% of the atoms before reaching the maximum number of iterations.

Besides the noiseless condition, we also made experiments in which the uniformly distributed positive noise of varying signal-to-noise ratios (SNRs) was corrupted to the test signals in order to evaluate the performance and robustness of antinoise. All trials were repeated 15 times with different initialized dictionaries. Figure 4 shows the results of the experiment for noise levels of 10, 20, and 30 dB and for the noiseless case. Obviously, NMFSC and NN-KSVD performed worst, especially under lower SNR conditions. NMFℓ^0-H performed better than NNSC, NMFSC, and NN-KSVD under various conditions. The proposed PCDDL performed best on dictionary learning, although it performed only slightly better than NMFℓ^0-H under various conditions. The average runtime of each trial for these algorithms was 35 s, 146 s, 244 s, 24 s, and 4 s, respectively. Obviously, PCDDL has a remarkable advantage in computational consumption. Note that, in the experiment, NN-KSVD and NMFℓ^0-H required a specified, exact number of nonzero elements in the coefficient matrix (3/50 = 0.06 for the case) as shown in Figure 3, and NMFSC was executed with a sparsity factor of 0.8 on the coefficients. For NNSC and PCDDL, the sparsity of the coefficient matrices was adjusted via the regularization parameters λ. In the experiment, the corresponding parameters λ were set to 0.2 in both the cases, which was calibrated off-line through several trials. The two parameters λ were fixed during iterations in order to reduce the number of iterations and computational cost.

4.2. Recovery Experiment of Decimal Digits Dictionary. To further investigate the potential practicality of the proposed PCDDL algorithm, we considered the 10 decimal digits dataset from [15]. The dataset is composed of 90 images of size 8×8, representing 10 decimal digits with various position shifts. Note that a mistake exists in the original dataset, in which some atoms are duplicated. In the original dataset, for example, the atoms of the first column are the same as the ones of the fifth column. Before the experiment, we corrected the problem by making all the atoms different.

Before beginning the experiment, we first generated 3000 training signals of size 64 × 1, each of which is a random linear combination of 5 different atoms with random positive coefficients. That is, there are uniformly 5 nonzero elements



(a)

(b)

Figure 1: Evolution of the rate of atom recovery versus the iteration number of five algorithms. (a) It shows 3000 iterations. (b) It is a close-up view of the former 200 iterations for (a).

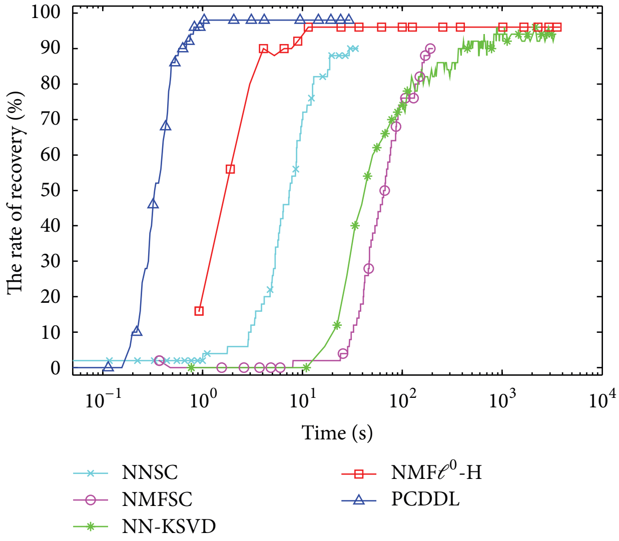


Figure 2: Evolution of the rate of atom recovery versus the runtime of five algorithms. These algorithms run 3000 iterations, respectively. PCDDL achieved the best rate of recovery in the least time.

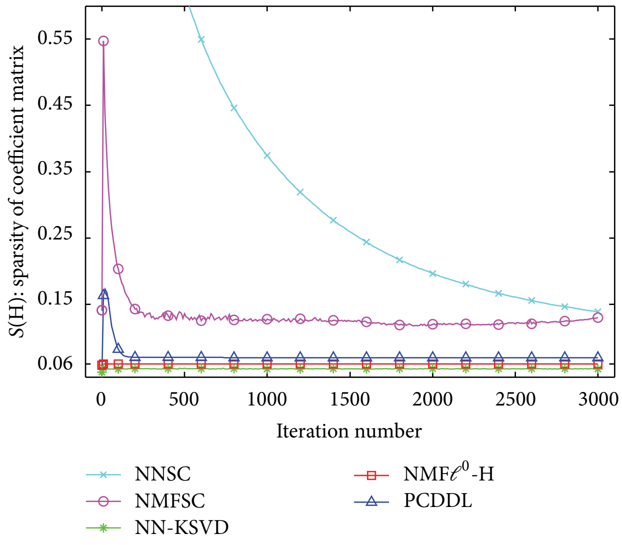


Figure 3: Evolution of the sparsity of the coefficient matrix versus the iteration number of five algorithms.

in each vector of the corresponding coefficient matrix. In order to learn original dictionary, the training signals were input into the five algorithms, NNSC, NMFSC, NN-KSVD, NMFℓ^0-H, and PCDDL. We also added the uniformly distributed positive noise of varying SNR to the training signals in order to evaluate the robustness of antinoise. The obtained results are shown in Figure 5.

As the results of the experiments in the above subsection, PCDDL performed better than the other four algorithms at three noise levels and in the noiseless case. The results of NN-KSVD were not as good as described in [15], because we corrected the above-mentioned mistake in the original dataset (i.e., removed duplicated atoms). The duplicated atoms in the original dataset led to the better, but wrong, result in [15] compared with the results of our experiment. Surprisingly, NNSC performed worst in this experiment, and it could almost not learn any correct atoms no matter how the parameters had been chosen. In a typical run, the average runtime of each trial was 412 s, 473 s, 822 s, 136 s and 23 s, respectively. This fact further shows that PCDDL has a remarkable advantage in computational consumption. In Figure 6, we give an example of the experiment under noiseless conditions, in which the four algorithms except

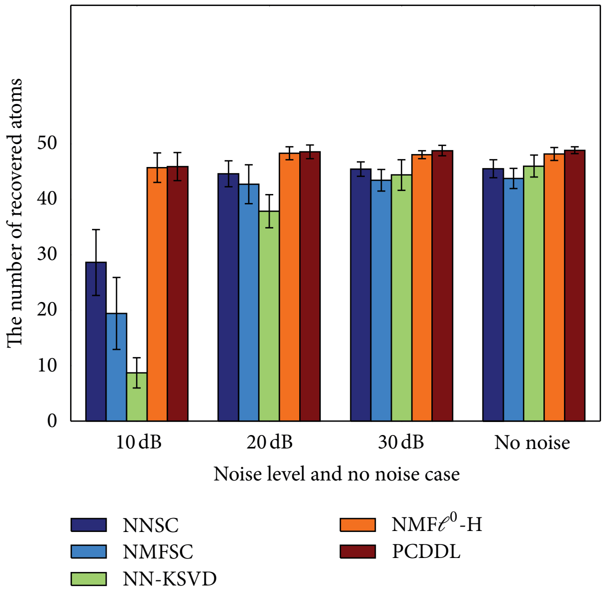


Figure 4: Results of a synthetic experiment with a dictionary of size 20 × 50. For each of the tested algorithms and for each noise level, 15 trials were performed. Averaged values of learned atoms and corresponding deviation values are displayed.

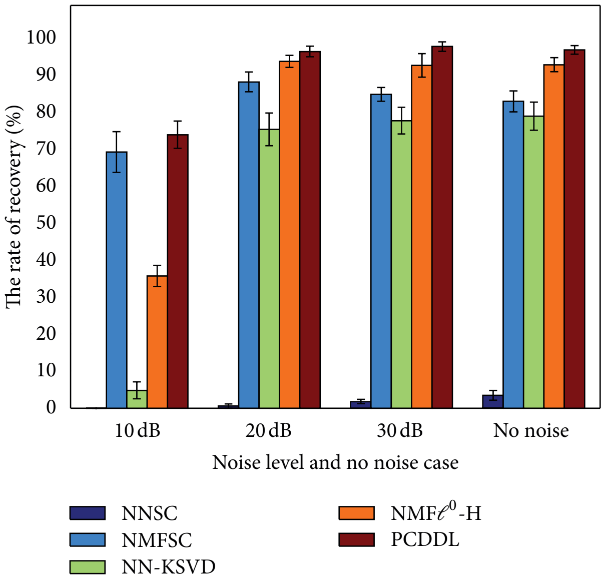


Figure 5: Results of a synthetic experiment with a decimal digits dictionary of size 64 × 90. For each of the tested algorithms and for each noise level, 10 trials were performed. Averaged values of learned atoms and corresponding deviation values are displayed.

NNSC recovered 77, 72, 86, and 89 atoms of 90 atoms, respectively. The result for NNSC was not showed in Figure 6 since it could almost not learn any correct atoms. Figure 6(a) shows the dataset revised by us. Figure 6(f) shows the result obtained by PCDDL, where only one digit 8 could not be recovered correctly. Certainly, PCDDL can either recover 100% of the atoms in considerable cases.

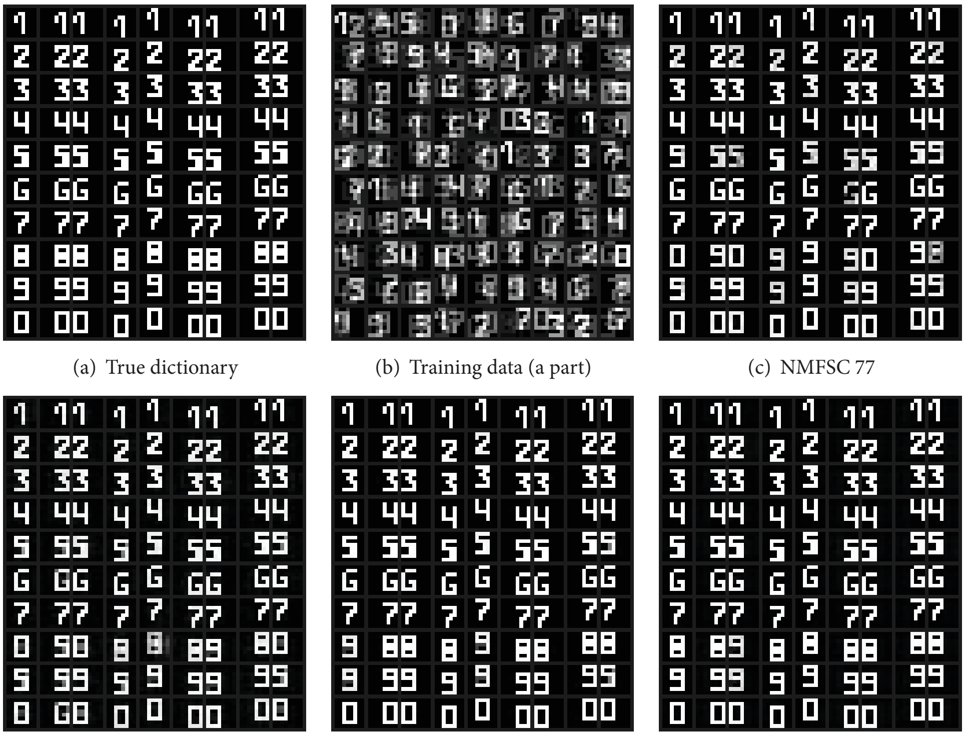
4.3. Image Denoising of Nature Images. Image denoising problem is important, not only because of the obvious applications that it serves. Being the simplest possible inverse problem, it provides a convenient platform through which image processing ideas and techniques can be assessed. In this sense, we intend to apply nonnegative dictionary learning to image denoising problem. Using redundant representations and sparsity as driving forces for denoising of signals constitutes significant progress [31, 32]. In these studies, a typical noise model is $\mathbf{Y} = \mathbf{X} + \mathbf{V}$, where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the clean image, $\mathbf{V} \in \mathbb{R}^{m\times n}$ is assumed to be white Gaussian noise with a fixed standard deviation σ (the case of nonuniform σ is dealt with in [33]), and $\mathbf{Y} \in \mathbb{R}^{m\times n}$ is the noisy observed image. Here, the noise is assumed to be uniformly distributed with nonnegative values, instead of zero-mean white and homogeneous Gaussian noise, since this paper is for studying the sparse representation of nonnegative signals. For solving the denoising problem, we adopted the algorithm presented in [31], which is based on a sparse and redundant representation model on small image patches. In the procedure, the original dictionary learning algorithm is replaced with our proposed PCDDL.

In this set of experiments, the dictionaries used were of size 64 × 256, which were designed to handle image patches of size 8 × 8 pixels. All reported results are presented as an average of three experiments, having different realizations of the noise. Some standard test images including Barbara (512 × 512), House (256 × 256), Boats (512 × 512), Lena (512 × 512), and Peppers (256 × 256) were used in the experiment. We added noise of various levels to the test images. We used two quality measures, the peak SNR (PSNR) and the structural similarity (SSIM), to assess the denoised images. Let $\mathbf{X}$ and $\widehat{\mathbf{X}}$ denote the ideal image and the deteriorated image, respectively. We calculate the PSNR value of $\widehat{\mathbf{X}}$ by $\mathrm{PSNR}(\widehat{\mathbf{X}}) = 10 \cdot \log_{10}(1/(\mathbf{X} - \widehat{\mathbf{X}})^2)$. For SSIM, its value range is between 0 and 1, and its value equals 1 if $\mathbf{X} = \widehat{\mathbf{X}}$. For more information about the SSIM index, please refer to references in [34].

In the experiment, we focused on tests with higher noise levels, because it may be more critical. We chose the conventional Wavelets denoising algorithm [35] and the known nonlocal means (NL-means) algorithm [36] as the compared objects. Additionally, we also chose the NMFℓ^0-H because of its better performance in previous experiments. It is notable that NMFℓ^0-H is very time-consuming for the dictionary learning procedure, as described in the two experiments above. Table 1 summarizes the results of the denoising experiment. We concluded that the denoising algorithm using the PCDDL dictionary achieved highly competitive PSNR and SSIM performance outcomes compared to that of Wavelets, NL-means, and NMFℓ^0-H algorithms. When comparing PSNR, the denoising algorithm using the PCDDL dictionary outperformed NL-means in the range of about 0.7 dB~2 dB and performed much better than the Wavelets and NMFℓ^0-H algorithms. When comparing the SSIM index, the denoising algorithm using the PCDDL dictionary returned results comparable to that of the NL-means algorithm. Subjective quality comparisons for two

Table 1: PSNR (dB) and SSIM results for different algorithms. In each cell, four groups of denoising results are shown. Top row, Wavelets; second row, NL-means; third row, NMFℓ^0-H; bottom row, PCDDL.

Input PSNR	Lena		Barbara		Boat		House		Pepper		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
29.97	32.91	0.8775	29.61	0.8633	30.28	0.8093	33.05	0.8669	31.20	0.8881	31.41	0.8610
	37.76	0.9370	36.62	0.9583	35.55	0.9205	38.17	0.9357	36.02	0.9485	36.82	0.9400
	29.23	0.9224	33.15	0.9538	35.95	**0.9451**	33.68	0.9519	35.76	0.9568	33.55	0.9460
	39.39	**0.9491**	**38.69**	**0.9641**	**38.08**	0.9445	**40.10**	**0.9580**	**38.63**	**0.9577**	**38.98**	**0.9547**
20.12	28.52	0.7982	25.03	0.7168	26.63	0.6988	28.03	0.7856	26.24	0.7902	26.89	0.7579
	33.45	0.8756	31.74	0.8914	31.04	0.8148	34.06	0.8729	32.06	**0.8929**	32.47	0.8695
	29.61	0.8680	30.82	**0.9056**	29.17	0.8232	33.93	**0.8846**	28.22	0.8788	30.35	**0.8720**
	34.39	**0.8791**	**32.58**	0.8870	**32.24**	**0.8376**	**34.32**	0.8714	**32.70**	0.8839	**33.25**	0.8718
14.09	25.74	0.7312	22.71	0.6044	24.23	0.6101	24.90	0.7220	23.15	0.7021	24.15	0.6740
	30.14	**0.8039**	27.52	**0.7867**	27.69	**0.7145**	30.44	0.8087	28.52	**0.8235**	28.86	**0.7875**
	27.68	0.7870	24.63	0.7411	24.10	0.6578	27.64	**0.8170**	22.87	0.7604	25.38	0.7527
	31.24	0.7953	**28.71**	0.7700	**28.80**	0.7137	**31.21**	0.7981	**29.27**	0.7933	**29.85**	0.7741
8.82	23.47	0.6712	21.07	0.5249	22.32	0.5384	22.51	0.6666	20.56	0.6164	21.99	0.6035
	27.18	**0.6876**	24.32	**0.6402**	24.99	**0.5938**	26.60	0.6767	24.97	**0.7114**	25.61	**0.6619**
	21.58	0.6677	19.60	0.5175	20.38	0.5190	21.41	0.6746	19.31	0.6232	20.46	0.6004
	28.38	0.6727	**25.26**	0.5986	**26.08**	0.5693	**28.37**	**0.6917**	**26.27**	0.6643	**26.87**	0.6393



(a) True dictionary

(b) Training data (a part)

(c) NMFSC 77

(d) NN-KSVD 72

(e) NMFℓ^0-H 86

(f) PCDDL 89

Figure 6: (a) True dictionary composed of 90 atoms. (b) Part of the total training data. (c)–(f) Learned dictionaries from NMFSC, NN-KSVD, NMFℓ^0-H, and PCDDL algorithms. The numbers of learned atoms are 77, 72, 86, and 89, respectively. Note that these resulting dictionaries have been realigned to facilitate comparison with the original dictionary.

typical test images (Boat and House) are shown in Figures 7 and 8. The PCDDL dictionary learned from the noisy House image in Figure 8 is illustrated in Figure 9.

4.4. Human Face Image Analysis. In this subsection, we describe our experiment on learning a global-based representation [21] using a face dataset. The learning process can be considered to be one kind of principal component analysis. We used the ORL dataset of faces (online available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html). Since the ORL dataset includes 400 facial images of size 92 × 112 pixels, the dataset can be considered to be large scale. Using the dataset, we can evaluate the computational performances of the PCDDL and the other

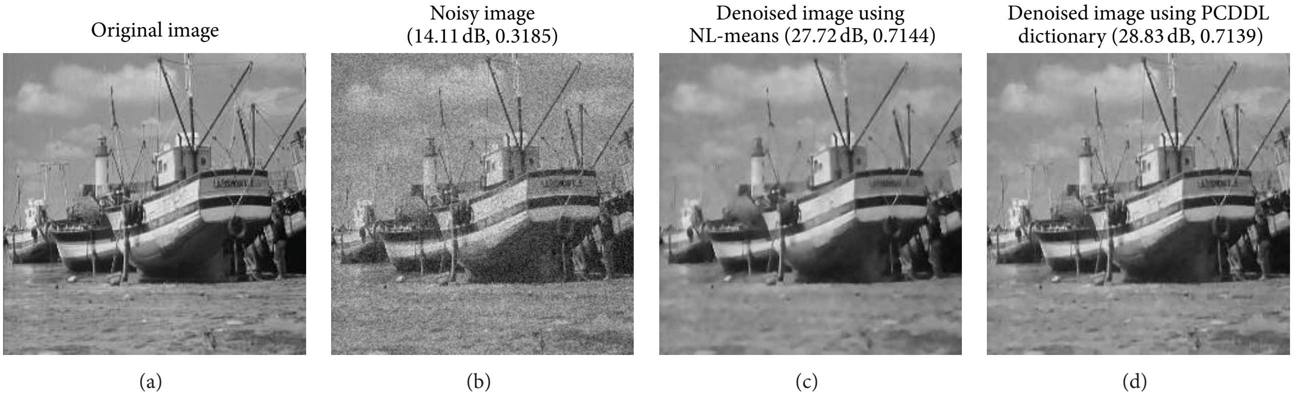

Original image
Noisy image (14.11 dB, 0.3185)
Denoised image using NL-means (27.72 dB, 0.7144)
Denoised image using PCDDL dictionary (28.83 dB, 0.7139)
(a)
(b)
(c)
(d)

Figure 7: Example of denoising results for the image "Boat" with a noise level of 14.11 dB. In brackets, the former items denote PSNR values, and the latter items denote the SSIM index.

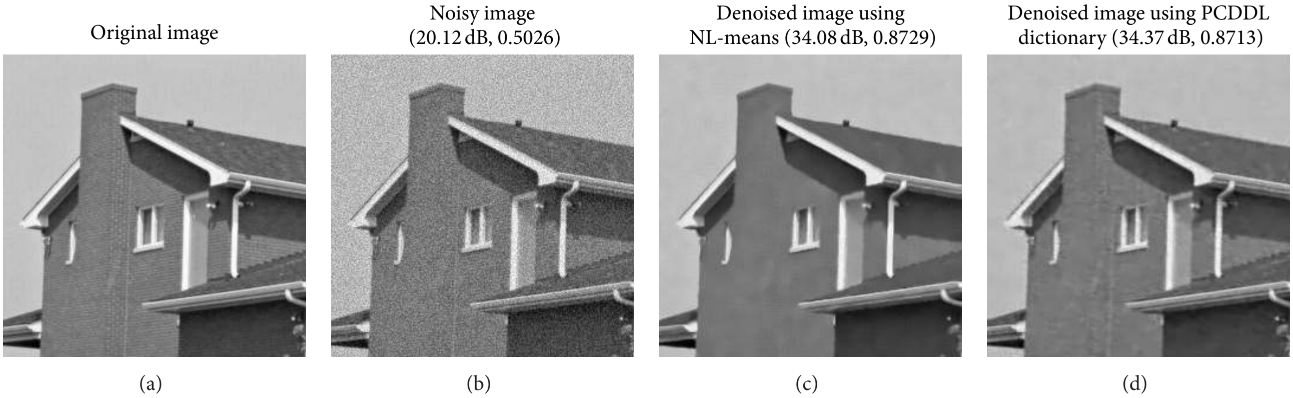

Original image
Noisy image (20.12 dB, 0.5026)
Denoised image using NL-means (34.08 dB, 0.8729)
Denoised image using PCDDL dictionary (34.37 dB, 0.8713)
(a)
(b)
(c)
(d)

Figure 8: Example of the denoising results for the image "House" with the noise level of 20.12 dB. In brackets, the former items denote PSNR values, and the latter items denote the SSIM index.

compared algorithms. To assess the experiment fairly, we drove the compared algorithms to obtain the corresponding coefficient matrices and forced them to reach as comparable level of sparsity as possible (based on ℓ^0-norm). By using the Hoyer's sparsity measure for a vector $\mathbf{x} \in \mathbb{R}^n$, defined as

$$\text{Sparsity}(\mathbf{x}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1/\|\mathbf{x}\|_2}{\sqrt{n} - 1} \in [0,1], \tag{13}$$

we compared the average sparsity of all column vectors in these coefficient matrices. Additionally, we computed the respective relative errors defined below and counted the respective runtime

$$\text{Relative Error} = \frac{\|\mathbf{Y} - \mathbf{WH}\|_F}{\|\mathbf{Y}\|_F}. \tag{14}$$

In the experiment, we performed a global-based feature learning of rank $r = 36$ and constrained the coefficient matrices to have a sparsity of about 0.08; that is, each facial image was required to be represented with three facial features ($36 \times 0.08 \approx 3$). Besides NMFSC and NMFℓ^0-H, we chose another sparse NMF algorithm (denoted as SNMF) [20] as the compared objective. Note that NN-KSVD was not included in this experiment, since it has exceedingly high computational consumption. Each of these algorithms required some initialization parameters and a limit on the number of its iterations. For SNMF, we allowed 3000 iterations; and for the parameter α, which is used to adjust sparsity, we chose 100. For NMFSC, we only constrained the

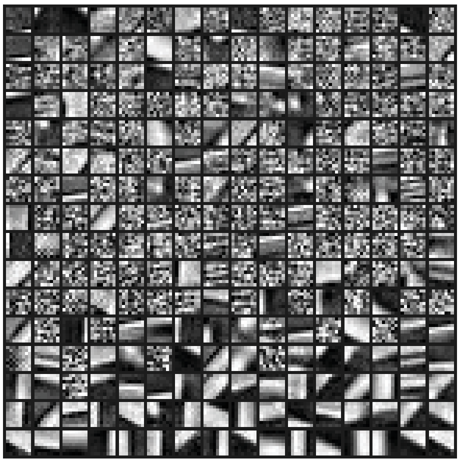


Figure 9: The PCDDL dictionary has a size of 64×256, which was learned from the noisy House image in Figure 8.

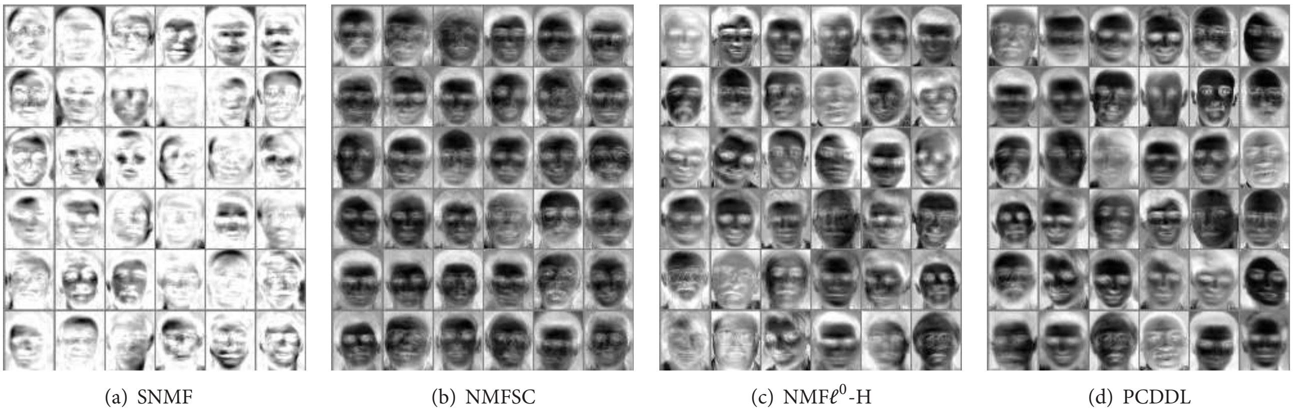

(a) SNMF (b) NMFSC (c) NMFℓ^0-H (d) PCDDL

Figure 10: Globally featured faces learned by SNMF, NMFSC, NMFℓ^0-H, and PCDDL.

sparsity of coefficient factor $\mathbf{H}$ to 0.9 in terms of (13) and executed at most 3000 iterations, which was necessary for convergence. For NMFℓ^0-H, we set the maximum number of nonzero elements of vectors in factor $\mathbf{H}$ to 3 ($3/36 \approx 0.0833$, close to 0.08) and allowed 30 iterations, considering the high computational consumption of NMFℓ^0-H. For the proposed PCDDL, we allowed at most 200 iterations, and λ was set to 10, that is, calibrated through several trials. All four algorithms were run three times with the same initial random matrices (for NMFℓ^0-H, it was not necessary to initialize coefficient $\mathbf{H}$). The averaged results are reported in Table 2.

Through Table 2, it can be observed that SNMF seems to be incapable of obtaining an actual sparse representation, despite the fact that it is designed to enhance sparsity by introducing the ℓ^1-norm. The other three algorithms obtained similar results and produced much sparser solutions, that is, more global-based representations. NMFSC and NMFℓ^0-H produced lower relative errors but took much more runtime than PCDDL. The runtime of NMFSC and NMFℓ^0-H was about 14 and 23 times longer than that of PCDDL. In view of its high efficiency, PCDDL is more suitable for large-scale data analysis. In Figure 10, we show an illustration of the global-based features learned by the four algorithms in a typical run.

Table 2: Comparisons of $S(\mathbf{H})$-based sparsity, Hoyer's sparsity (based on (13)), relative error (based on (14)), and runtime for SNMF, NMFSC, NMFℓ^0-H, and PCDDL.

Algorithm	$S(\mathbf{H})$	Sparsity	Relative error	Time (s)
SNMF	96.65	0.4314	0.9904	940
NMFSC	8.00	0.9490	0.2520	415
NMFℓ^0-H	8.33	0.8957	0.1852	662
PCDDL	8.00	0.9447	0.2925	28

5. Conclusion

In this paper, we presented a novel and efficient method for learning nonnegative dictionaries for sparse representation of nonnegative signals. In this method, we generalized the coordinate descent strategy for optimization for being able to be applied to a multivariable case, so that it can process in a parallel way. By this strategy we developed an efficient algorithm, which has been named as the parallel coordinate descent dictionary learning (i.e., PCDDL) algorithm. The algorithm updates the dictionary in a column-wise manner and the coefficient matrix in a row-wise manner. In each column-wise or row-wise updating, PCDDL optimizes a series of optimal problems sequentially, each of which is an optimization of a quadratic function. Furthermore, such optimization problems can be solved explicitly, so that the algorithm can be processed very precisely and quickly from a global perspective according to the properties of the univariate quadratic problem. For this reason, the proposed algorithm can efficiently solve the nonnegative dictionary learning problem with very high accuracy.

Results of experiments on dictionary recovery showed that PCDDL can correctly learn a nonnegative, overcomplete dictionary, regardless of wether the objective signals are synthetic data or are natural images. Additionally, further experiments supported the potential application of PCDDL in the field of image processing, such as image denoising, image classification, and large-scale data processing due to its low computational consumption. We are currently working on applying this method to some practical problems in image processing, for example, large-scale image classification. The results from these ongoing studies will be presented in the future.

References

[1] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications*, vol. 416, no. 1, pp. 29–47, 2006.

[2] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 765–777, 2007.

[3] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. 207–212, 2001.

[4] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol. 2, pp. 588–595, 2007.

[5] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*, vol. 42, pp. 373–386, 2006.

[6] M. Wang, W. Xu, and A. Tang, "A unique "nonnegative" solution to an underdetermined system: from vectors to matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1007–1016, 2011.

[7] M. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. Davies, "Sparse representations in audio and music: from coding to source separation," *Proceedings of the IEEE*, pp. 995–1005, 2010.

[8] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, pp. 972–982, 2010.

[9] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transaction on Signal Processing*, vol. 54, pp. 4311–4322, 2006.

[12] K. Engan, K. Skretting, and J. H. Husoy, "Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digital Signal Processing*, vol. 17, pp. 32–49, 2007.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[14] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[15] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD and its non-negative variant for dictionary design," in *Proceedings of the SPIE Conference Wavelets*, pp. 327–339.

[16] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[17] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 556–562, 2000.

[18] P. O. Hoyer, "Non-negative sparse coding," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565.

[19] J. Eggert and E. Korner, "Sparse coding and NMF," in *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 2529–2533.

[20] W. Liu, N. Zheng, and X. Lu, "Non-negative matrix factorization for visual coding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, pp. 293–296, 2003.

[21] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[22] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, pp. 452–456, SIAM, Philadelphia, Pa, USA, 2004.

[23] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, pp. 3970–3975, 2005.

[24] R. Peharz, M. Stark, and F. Pernkopf, "Sparse nonnegative matrix factorization using ℓ^0-constraints," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP '10)*, pp. 83–88, 2010.

[25] R. Peharz and F. Pernkopf, "Sparse nonnegative matrix factorization using ℓ^0-constraints," *Neurocomputing*, vol. 80, pp. 38–46, 2012.

[26] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 403–415, 2006.

[27] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, pp. 1495–1502, 2007.

[28] R. Tandon and S. Sra, "Sparse nonnegative matrix approximation: new formulations and algorithms," Tech. Rep. 193, MPI, 2010.

[29] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[30] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss-Seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, 2000.

[31] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

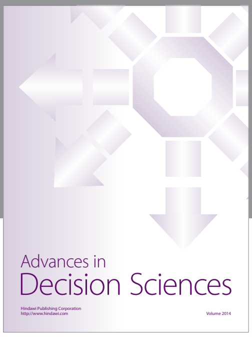
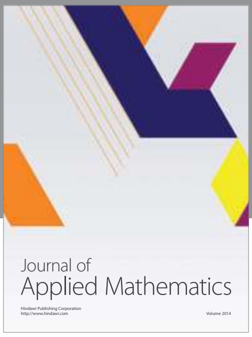
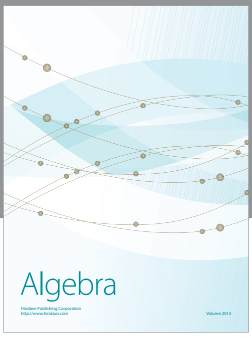
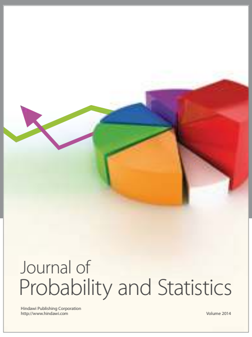
[32] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 457–464, 2011.

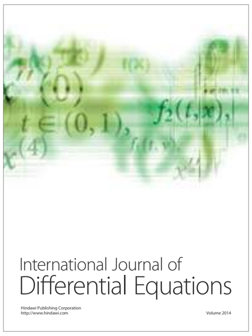
[33] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

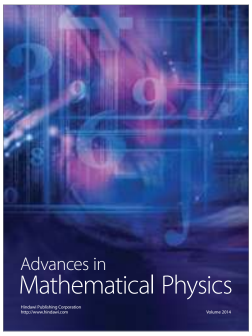
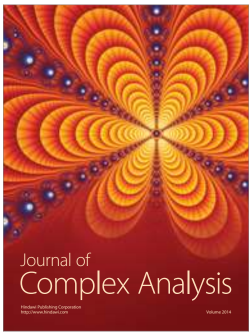
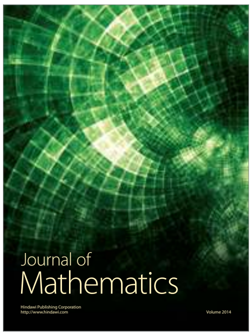
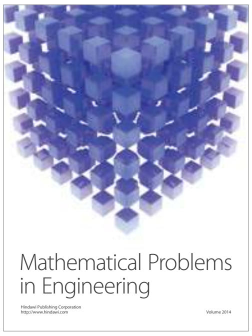
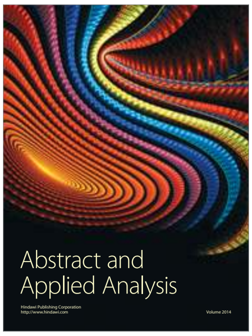
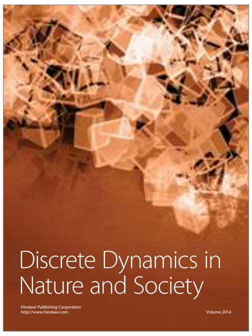
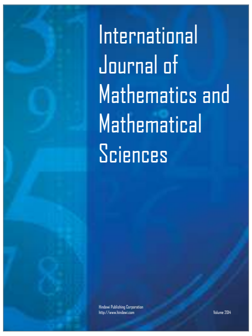
[34] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[35] R. Baraniuk, H. Choi, R. Neelamani, and V. Ribeiro, "Rice Wavelet Toolbox," 2011, http://dsp.rice.edu/software/rice-wavelet-toolbox/.

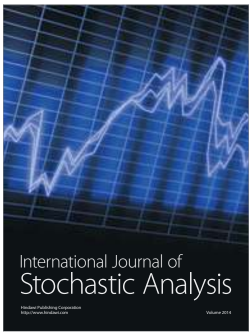
[36] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 60–65, 2005.

Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

Algebra

Journal of
Probability and Statistics

The Scientific
World Journal

International Journal of
Differential Equations

International Journal of
Combinatorics

Advances in
Mathematical Physics

Journal of
Complex Analysis

Journal of
Mathematics

Mathematical Problems
in Engineering

Abstract and
Applied Analysis

Discrete Dynamics in
Nature and Society

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces

International Journal of
Stochastic Analysis

Journal of
Optimization