PAPER

Dictionary Learning with Incoherence and Sparsity Constraints for Sparse Representation of Nonnegative Signals

Zunyi TANG^{†a)}, *Nonmember* and Shuxue DING^{††b)}, *Member*

SUMMARY This paper presents a method for learning an overcomplete, nonnegative dictionary and for obtaining the corresponding coefficients so that a group of nonnegative signals can be sparsely represented by them. This is accomplished by posing the learning as a problem of nonnegative matrix factorization (NMF) with maximization of the incoherence of the dictionary and of the sparsity of coefficients. By incorporating a dictionary-incoherence penalty and a sparsity penalty in the NMF formulation and then adopting a hierarchically alternating optimization strategy, we show that the problem can be cast as two sequential optimal problems of quadratic functions. Each optimal problem can be solved explicitly so that the whole problem can be efficiently solved, which leads to the proposed algorithm, i.e., sparse hierarchical alternating least squares (SHALS). The SHALS algorithm is structured by iteratively solving the two optimal problems, corresponding to the learning process of the dictionary and to the estimating process of the coefficients for reconstructing the signals. Numerical experiments demonstrate that the new algorithm performs better than the nonnegative K-SVD (NN-KSVD) algorithm and several other famous algorithms, and its computational cost is remarkably lower than the compared algorithms.

key words: dictionary learning, sparse representation, nonnegative matrix factorization (NMF), hierarchical alternating least squares (HALS), overcomplete dictionary, quadratic function

1. Introduction

Sparse representation of signals has recently received much attention as it has produced some promising results in various applications. However, to date, there are only few reports on sparse representation for nonnegative signals. Such problems are frequently encountered in signal and image processing [1], in handling of multispectral data [2], in considering nonnegative factorization for recognition [3], [4], and other important problems [5]. In this paper, we mainly focus on sparse representation for nonnegative signals.

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, or words, 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 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 pre-specified set of functions or learned by a given set of training signals. In practice [6], a learned dictionary has proved critical for achieving superior results in the domains of signal and image processing. In some sense, one can discover and properly understand the crucial causes underlying the sensed signals by the learned dictionary. Therefore, how to learn a dictionary is important for sparse representation.

Naturally, the problem of finding a dictionary and its sparse representation with the fewest number of atoms can be modeled by using the minimization of ℓ^0 -norm upon the column vectors of $\mathbf{H}$. Considering the fact that the ℓ^0 -norm optimization problem is generally NP-hard, one frequently used heuristic is the ℓ^1 -minimization [7], which returns the vector with the least ℓ^1 -norm. A series of studies has led to many dictionary learning algorithms such as LARS [8], K-SVD [9], ILS-DLA [10], and RLS-DLA [11]. 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” [12], 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) [1], [2], [13] has been widely applied to data analysis having nonnegativity constraints. NMF aims to 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 algorithm [14] does not impose any constraints on the two factors except for the nonnegativity, and so it is not sufficient to lead to a sparse enough representation. In order to obtain a sparser representation, various sparsity constrained NMF methods have been proposed. Hoyer *et al.* [15], [16] considered enforcing the sparsity of coefficient matrix using ℓ^1 -norm. Hoyer [17] also introduced a measure of sparsity based on the ratio of the ℓ^1 -norm of a vector to the ℓ^2 -norm. Some authors imposed sparsity constraints using ℓ^2 -norm [18], [19]. Peharz *et al.* [20], [21] presented sparse NMF methods that constrain the ℓ^0 -(pseudo)-norm of the coefficient matrix.

In this paper, we present a new NMF-based method for learning nonnegative, overcomplete dictionaries for sparse

Manuscript received November 28, 2012.

Manuscript revised January 27, 2013.

[†]The author is with Graduate School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu-shi, 965-8580 Japan

^{††}The author is with the Faculty of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu-shi, 965-8580 Japan

a) E-mail: d8111103@u-aizu.ac.jp

b) E-mail: sdng@u-aizu.ac.jp

DOI: 10.1587/transinf.E96.D.1192

representation of nonnegative signals. In this method, we force the representation coefficients to be as sparse as possible and simultaneously force the atoms of the dictionary to be as incoherent as possible for obtaining sparser representations. For optimizing our objective function, we propose optimizing each of two factors alternately and separating each optimization into column-wise or row-wise suboptimization problems, which is meant by the terms “alternating” and “hierarchical”. In each column-wise or row-wise updating, one can optimize a series of optimal problems sequentially, each of which is an optimization of a quadratic function that can be solved explicitly, so that the whole problem can be efficiently solved. The way of optimizing each subproblem is obviously different from the standard hierarchical alternating least squares (HALS) [1], which is an efficient method for solving the NMF problem. This optimization method may be regarded as a variation of HALS. Therefore, our proposed algorithm is termed as the sparse HALS (SHALS). Through experimental validations, we observed that our algorithm achieves the best rate of atom recovery compared to the conventional algorithms [12], [15], [17], [21]. 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 inherent to factorization.

The remaining part of the paper is organized as follows. In Sect. 2, we formulate the nonnegative dictionary learning problem and review several existing algorithms in the literature. In Sect. 3, we describe our proposed SHALS algorithm in detail. In Sect. 4, we report the results of numerical experiments using SHALS and compare these results with those of several other existing algorithms. Finally, in Sect. 5, we draw our conclusions and discuss related research topics for the future.

2. Problem Formulation and Existing Methods

2.1 Nonnegative Dictionary Learning Problem

In general, a formal description of dictionary learning is as follows. Given an input signal matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$, dictionary learning is to build or learn a dictionary of words, $\mathbf{W} \in \mathbb{R}^{m \times r}$ ($r > m$), by which the signal $\mathbf{Y}$ can be sparsely represented. That is to say, based on the learned dictionary $\mathbf{W}$, we can find a sparse coefficient matrix $\mathbf{H} \in \mathbb{R}^{r \times n}$ such that

$$\mathbf{Y} = \mathbf{W}\mathbf{H} \quad \text{or} \quad \mathbf{Y} \approx \mathbf{W}\mathbf{H} \quad (1)$$

where the sparsity of $\mathbf{H}$ can be defined as $S(\mathbf{H}) = \frac{\|\mathbf{H}\|_0}{r \times n}$, which calculates the ratio of the number of nonzero elements and the number of all elements in $\mathbf{H}$. If $\mathbf{Y}$, $\mathbf{W}$, and $\mathbf{H}$ are all limited to be nonnegative, then the process is called nonnegative dictionary learning. Note that $r > m$ means that dictionary is overcomplete. If $r \ll \min(m, n)$, the dictionary is overdetermined and often used for feature extraction.

2.2 Existing Methods

General sparsity constrained NMF methods can be used for nonnegative dictionary learning. Hoyer [15] proposed nonnegative sparse coding (NNSC), which minimizes the cost function $\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_F^2 + \lambda \sum_{i,j} \mathbf{H}_{ij}$, and is subject to nonnegativity constraints on $\mathbf{W}$ and $\mathbf{H}$. Liu *et al.* [16] presented sparse NMF (SNMF) similar to NNSC. In SNMF the Kullback-Leibler divergence term is adopted to quantify the quality of the approximation of the representation instead of using a Euclidean-type function. Furthermore, Hoyer [17] proposed another sparsity constrained NMF method, termed NMFSC. The method closely follows the standard NMF method proposed by Lee and Seung [14]. This is done by imposing sparsity constraints on both the dictionary matrix and the coefficient matrix, or either of the two. Note that in many steps of NMFSC, sparse projector operations are included for enforcing the sparsity of factors $\mathbf{W}$ and $\mathbf{H}$, by explicitly setting the ℓ^1 -norm and ℓ^2 -norm of each column or row in the factors. These lead to the heavy computational cost of NMFSC.

A more recent method related to the sparsity constrained NMF has been proposed by Peharz *et al.* [20], [21]. Inspired by the active-set algorithm for nonnegative least squares (NNLS), the authors proposed a nonnegative, sparse coding algorithm, the so-called reverse sparse NNLS (rsNNLS). As stated in [21], this method can be regarded to impose an ℓ^0 -pseudo-norm sparsity constraint on the coefficient vector. Subsequently, by combining the rsNNLS with a generic, unconstrained alternating nonnegative least squares (ANLS), the authors constructed a sparsity constrained NMF method, i.e., NMF^{ℓ^0} -H. This method is still an iterative two-stage algorithm. Note that for NMF^{ℓ^0} -H, *a priori*, fixed sparsity for the coefficient matrix $\mathbf{H}$ is required.

Aharon and Elad [9] proposed an appealing dictionary learning algorithm named K-SVD. To make K-SVD suitable for the nonnegative dictionary learning, the authors derived a nonnegative variation of K-SVD, NN-KSVD, based on two main modifications on K-SVD. One uses a nonnegative variation of basis pursuit (NNBP) [22] instead of orthogonal matching pursuit (OMP) [23] in the sparse coding stage. The other modification is involved in the dictionary update stage, where negative values generated after each SVD iteration are simply truncated to zero in order to maintain nonnegativity in the dictionary. However, the latter *ad hoc* change leads to the obvious performance reduction of NN-KSVD compared with K-SVD. Furthermore, NN-KSVD needs *a priori* sparsity of coefficient, as it is for NMF^{ℓ^0} -H.

3. Nonnegative Dictionary Learning with Incoherence and Sparsity Constraints

3.1 On the Incoherence of Dictionary $\mathbf{W}$

If a dictionary is more efficient, the redundancy between

words should be smaller. Therefore, one should enhance the incoherence of dictionary during dictionary learning. As stated in [7], [24], the uniqueness of the learned dictionary depends not only on the sparsity of the coefficient matrix $\mathbf{H}$ but also on the incoherence of the dictionary matrix $\mathbf{W}$. Therefore, we propose incorporating the following penalty term to enhance the incoherence of $\mathbf{W}$:

$$\sum_{i=1}^r \sum_{j=1, i \neq j}^r \mathbf{w}_i^T \mathbf{w}_j \quad (2)$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ are different column vectors in $\mathbf{W}$. If each column of $\mathbf{W}$ is normalized to the unit ℓ^2 -norm, the penalty term (2) is equivalent to $\sum_{i=1}^r \sum_{j=1, i \neq j}^r \cos \varphi_{i,j}$, which sums the cosine values of angles between any two different vectors. Since each vector is confined in the first quadrant because of its nonnegativity, the cosine values of the angles are always nonnegative. Hence, reducing the penalty term (2) means enlarging angles between vectors. This also means decreasing redundancies between vectors and enhancing the incoherence of dictionary. Actually, (2) can be viewed as a simplification of $\|\mathbf{W}^T \mathbf{W}\|_F^2$ [25].

3.2 Formulation and Algorithm

To enhance the sparsity of coefficients $\mathbf{H}$ and the incoherence of dictionary $\mathbf{W}$, we formulate the following sparsity-incoherence penalized NMF optimization problem:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{H}} f &= \|\mathbf{Y} - \mathbf{WH}\|_F^2 + \alpha \|\mathbf{H}\|_1 + \beta \sum_{i=1}^r \sum_{j=1, i \neq j}^r \mathbf{w}_i^T \mathbf{w}_j \\ \text{subject to } & \mathbf{W} \geq 0, \mathbf{H} \geq 0 \end{aligned} \quad (3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\mathbf{H}\|_1$ means $\sum_{i,j} |\mathbf{H}_{ij}|$. $\alpha > 0$ is a regularization parameter to suppress $\|\mathbf{H}\|_1$, which is utilized to control the sparsity of $\mathbf{H}$. $\beta > 0$ is a regularization parameter, which is used to adjust the incoherence of $\mathbf{W}$.

To solve the objective problem (3), we employ hierarchically alternating update strategy; i.e., updating one of two factors while fixing the other and optimizing each factor hierarchically rather than wholly optimizing as in the standard NMF method [14]. Furthermore, differently from the gradient descent process used in the standard HALS [1], we found that (3) 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. The detailed derivations of the update rules are given in Appendix A, and the final update rules are as follows:

$$\mathbf{W}_{:k}^* = \max \left(0, \frac{\mathbf{R}_k \mathbf{H}_{:,k}^T - \mu \sum_{l=1, l \neq k}^r \mathbf{W}_{:,l}^T \mathbf{H}_{:,k}^T}{\|\mathbf{H}_{:,k}\|_2^2} \right) \quad (4)$$

$$\mathbf{H}_{k,:}^* = \max \left(0, \frac{\mathbf{W}_{:,k}^T \mathbf{R}_k - \lambda}{\|\mathbf{W}_{:,k}\|_2^2} \right) \quad (5)$$

where $\mathbf{H}_{k,:}$ and $\mathbf{W}_{:,k}$ denote the k -th row of matrix $\mathbf{H}$ and the k -th column of matrix $\mathbf{W}$, respectively. $\mathbf{R}_k = \mathbf{Y} - \sum_{l=1, l \neq k}^r \mathbf{W}_{:,l} \mathbf{H}_{:,l}$, $\mu = \frac{\beta}{2}$, and $\lambda = \frac{\alpha}{2}$. Note that if $\mu = \lambda = 0$, then (4) and (5) degenerate into the update rules for the standard HALS [1] although they are derived with different optimization methods. In this viewpoint, our algorithm can be regarded as a generalization of HALS, thus being termed as sparse HALS (SHALS). Note that the way of maintaining the nonnegativity of two factor matrices in the SHALS 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 (see Appendix A). 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.

Remark 1: Although the two penalty terms are incorporated into the NMF formulation, the objective problem can still be optimized alternately and hierarchically. Furthermore, it can be cast as two sequential optimal problems of quadratic functions, each of which has a closed-form solution (see Appendix A).

Remark 2: The incoherence of $\mathbf{W}$ and the sparsity of $\mathbf{H}$ can be flexibly controlled by adjusting two regularization parameters of penalty terms.

Remark 3: SHALS 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.

3.3 Determinations of Parameters and Summary of SHALS Algorithm

In the step of updating $\mathbf{W}$ with a fixed $\mathbf{H}$, the parameter $\mu > 0$ can be adjusted for a tradeoff between the accuracy of reconstruction and the incoherence of $\mathbf{W}$. In terms of (4), μ should be less than $\|\mathbf{U}_{:,k}\|_\infty$ where $\mathbf{U}_{:,k} = \mathbf{R}_k \mathbf{H}_{:,k}^T / \sum_{l=1, l \neq k}^r \mathbf{W}_{:,l}^T \mathbf{H}_{:,k}^T$ ($/$ denotes element-wise division), otherwise $\mathbf{W}_{:,k}$ will become a zero vector that should be avoided in the course of dictionary learning. For μ , the closer it is to $\|\mathbf{U}_{:,k}\|_\infty$, the more difficult it is to obtain a sparse enough coefficient matrix. We further found that it is a better way to set μ to $0.01 \times \|\mathbf{U}_{:,k}\|_\infty$ because one can obtain relatively low approximation error and reduce adequately the incoherence of dictionary, and can also improve the degree of sparsity of the coefficient matrix.

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 $\|\mathbf{Y} - \mathbf{WH}\|_F^2$ and the sparsity of coefficient matrix $\mathbf{H}$, and plays an important role in the proposed method. To steer the solution toward a global, optimal solution, the parameter λ can be determined by two kinds of ways, off-line calibrating and adaptively 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.

Fig. 1 Plot of angles between any two atoms in the dictionaries, which are learned by the objective functions with and without dictionary-incoherence penalty term, respectively.

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 (5), otherwise $\mathbf{H}_k$ will become a zero vector. We may initialize λ with a very small value; e.g., 0.001, which can generally satisfy the above condition. Next, we alternately update $\mathbf{W}$ and $\mathbf{H}$ in terms of (4) and (5), and adjust λ according to the rule defined as follows,

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

where $\lambda^{(k)}$ and $S(\mathbf{H}^{(k)})$ denote the value of λ and the sparsity of $\mathbf{H}$ in the k -th iteration, respectively. S^* denotes the expected or *a priori* 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.

The proposed SHALS algorithm is devised to obtain a sparse enough representation while reducing the coherence of the learned dictionary. Therefore, one way to show the performance of the SHALS algorithm is to compare the changes of angles between any two atoms in the two different dictionaries, which are learned by the SHALS ($\mu \neq 0$) and the SHALS with the only sparsity penalty term ($\mu = 0$), respectively. Through the two algorithms, we achieved two groups of dictionary learning results from a nature image (“House” in Fig. 6), and we plotted in Fig. 1 the sorted angles between atoms in the learned dictionaries. It can be seen that the angles between atoms in the dictionary learned by SHALS ($\mu \neq 0$) are enlarged and the incoherence of dictionary is obviously enhanced.

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

Algorithm 1 SHALS

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 λ ;

- 1: **while** stopping criterion not satisfied **do**
- 2: Computing $\mathbf{P} = \mathbf{YH}^T$ and $\mathbf{Q} = \mathbf{HH}^T$;
- 3: **for** $k = 1$ to r **do**
- 4: $\mu \leftarrow 0.01 \times \max \left(\frac{\mathbf{P}_{:k} - \sum_{l=1, l \neq k}^r \mathbf{W}_{:l} \mathbf{Q}_{lk}}{\sum_{l=1, l \neq k}^r \mathbf{W}_{:l}^T} \right)$
- 5: $\mathbf{W}_{:k} \leftarrow \max \left(0, \frac{\mathbf{P}_{:k} - \sum_{l=1, l \neq k}^r \mathbf{W}_{:l} \mathbf{Q}_{lk} - \mu \sum_{l=1, l \neq k}^r \mathbf{W}_{:l}^T}{\mathbf{Q}_{kk}} \right)$
- 6: Normalizing $\mathbf{W}_{:k} \leftarrow \frac{\mathbf{W}_{:k}}{\|\mathbf{W}_{:k}\|_2}$
- 7: **end for**
- 8: Computing $\mathbf{U} = \mathbf{W}^T \mathbf{Y}$ and $\mathbf{V} = \mathbf{W}^T \mathbf{W}$;
- 9: **for** $k = 1$ to r **do**
- 10: $\mathbf{H}_{:k} \leftarrow \max \left(0, \frac{\mathbf{U}_{:k} - \sum_{l=1, l \neq k}^r \mathbf{V}_{kl} \mathbf{H}_{:l}}{\mathbf{V}_{kk}} \right)$
- 11: **end for**
- 12: Using the fixed λ or adaptively tuning λ according to the change of the sparsity of $\mathbf{H}$;
- 13: **end while**

3.4 Convergence Analysis of SHALS Algorithm

The standard NMF algorithm [14] belongs 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. Grippio and Sciandrone analyzed the convergence of the two-block convex optimization problems in [26]. They demonstrated that under the condition of the objective function differentiable continuously, a two-block convex optimization method 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, SHALS is such a two-block convex optimization algorithm, so that we can make analysis of its convergence by using the facts in [26]. During iterations, SHALS 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 $\|\mathbf{H}\|_1$ in (3) 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 $\alpha > 0$ and $\beta > 0$, (3) is differentiable with respect to $\mathbf{W}$ and $\mathbf{H}$, respectively. The existence of limit points and the differentiability of the objective function in (3) imply that the assumptions of Grippio and Sciandrone’s Corollary [26] are satisfied so that we can establish that the two-block minimization processes of SHALS converge.

4. Numerical Experiments

In this section, first we present the results of two experiments using SHALS with synthetic signals. The aims of the experiment are (1) to test whether the SHALS algorithm can recover the true dictionary, which was used to generate the test data; and (2) to compare the results with those of other algorithms such as NNSC[†] [15], NN-KSVD^{††} [12],

[†] Available at: <http://www.cs.helsinki.fi/u/phoyer/>

^{††} Available at: <http://www.cs.technion.ac.il/~elad/>

NMFSC[†] [17], and NMF ℓ^0 -H^{††} [21]. Next, we apply SHALS to a conventional digital image processing problem, image denoising, to verify the applicability of the proposed algorithm to real-world signals. Finally, we make an experiment of learning a global-based representation on a face dataset in order to demonstrate the applicability 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 SHALS algorithm for a nonnegative dictionary, we made 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, and runtime, *etc.*, we assess the methods under consideration (see above). The processes are as follows. We first generated a stochastic nonnegative matrix of size 20×50 with i.i.d. uniformly distributed entries, as described in [9]. Each vector was normalized to the 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, NMF ℓ^0 -H, and SHALS on the test signals. For the five algorithms, the initialized dictionary matrix of size 20×50 was composed of the randomly selected parts of the test signals. For NNSC, NMFSC, and SHALS, the corresponding coefficient matrices were initialized with i.i.d. uniformly distributed random nonnegative entries. NN-KSVD and NMF ℓ^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 [9], 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 Figs. 2, 3, and 4. NNSC and NMFSC took, respectively, about 3000 iterations to reach convergence while NMF ℓ^0 -H took only dozens of iterations. In addition, we also consider the runtime of each algorithm as showed in Fig. 4. Thus, we set the maximum numbers of iterations for NNSC, NMFSC, NN-KSVD, NMF ℓ^0 -H, and SHALS to 3000, 3000, 300, 30, and 500, respectively. Cer-

Fig. 2 Evolution of the sparsity of the coefficient matrix versus the iteration number of six algorithms. HALS has been added to this experiment to demonstrate that it cannot obtain a coefficient matrix with high sparsity due to lack of a sparsity constraint on coefficient matrix.

Fig. 3 Evolution of the rate of atom recovery versus the iteration number of five algorithms. Top: it shows 3000 iterations. Bottom: it is a close-up view of the former 200 iterations for the top figure.

tainly, 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.

[†] Available at: <http://www.cs.helsinki.fi/u/phoyer/>

^{††} Available at: <http://www.spac.tugraz.at/tools/>

Fig. 4 Evolution of the rate of atom recovery versus the runtime of five algorithms. These algorithms run 3000 iterations, respectively. SHALS achieved the best rate of recovery in the least time.

Fig. 5 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.

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 anti-noise. All trials were repeated 15 times with different initialized dictionaries $\mathbf{W}$. Figure 5 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 the worst, especially under lower SNR conditions. NMF^{ℓ^0} -H performed better than NNSC, NMFSC, and NN-KSVD under various conditions. The proposed SHALS performed best on dictionary learning, although the differences between it and NMF^{ℓ^0} -H under various conditions are small. The average runtime of each trial for these algorithms was 35 s, 146 s, 244 s, 24 s, and 4 s, respectively. Obviously, SHALS has a remarkable advantage in computational cost. Note, in the experiment, NN-KSVD and NMF^{ℓ^0} -H required a specified, exact number of non-zero elements in the coefficient

Table 1 Comparison of the rate of dictionary recovery (%) under various pre-specified sparsity values and various SNR level conditions. Note that sparsity decreases when the number of nonzero components increases. “N” denotes the number of nonzero components of the column vectors in coefficient matrices. “T” denotes the average runtime of trials.

Method	N	10 dB	20 dB	30 dB	No Noise	T(s)
NNSC	5	60.7	92.5	93.5	92.7	493
	10	0	43.8	61.5	74.3	496
	20	0	0	0	0	502
	(3-12)	2.3	77.7	87.5	88.3	497
NMFSC	5	0	38.6	43.3	51.9	1149
	10	0	0	0	0	1128
	20	0	0	0	0	1097
	(3-12)	0	0	0	0	1181
NN-KSVD	5	3.2	30.7	45.1	61.8	1145
	10	0	0	0	1.5	1842
	20	0	0	0	0	3800
	(3-12)	-	-	-	-	-
NMF^{ℓ^0} -H	5	96.2	98.4	98.7	98.6	257
	10	0	98.3	98.9	99.2	526
	20	0	0	0	0	954
	(3-12)	-	-	-	-	-
SHALS	5	96.2	98.5	99.2	98.7	144
	10	5.2	96.8	98.3	99.1	146
	20	0	50.2	86.8	94.3	145
	(3-12)	84.3	94.5	97.3	98.7	145

matrix ($3/50=0.06$ for the case) as shown in Fig. 2, and NMFSC was executed with a sparsity factor of 0.8 on the coefficients. For NNSC and SHALS, the sparsity of the coefficient matrix were adjusted via the regularization parameters λ . In the experiment, the corresponding parameters λ were set to 0.2 in both the cases, which is calibrated off-line through several trials. The two parameters λ are fixed during iterations in order to reduce the number of iterations and computational cost.

We repeated this experiment, in which the true dictionary $\mathbf{W}$, with a size 100×200 , was still generated with i.i.d. uniformly distributed entries and was normalized to the unit ℓ^2 -norm. We then synthesized four groups of 3000 test signals of dimension 100, each of which was produced by 5, 10, 20, and random number (between 3 and 12) of atoms, respectively, with the corresponding coefficients in random and independent positions. In the experiment, we also provided the corresponding averaged runtime of each trail for these algorithms.

Table 1 shows the recovery rates of the dictionaries achieved by these algorithms, which correspond to different coefficients of varying sparsity under different noisy and noiseless conditions. NMFSC and NN-KSVD performed the worst, with only a small portion of dictionary atoms being recovered if the sparsity was high. The corresponding results were worse at lower SNR values. NMF^{ℓ^0} -H produced better results than NNSC, NMFSC, and NN-KSVD under conditions of exactly fixed and assigned sparsity, but it seemed to be incapable of learning with a relatively less sparse coefficient; for example, when the number of nonzero coefficients was 20. NNSC, on the other hand, did not require *a priori* sparsity of the coefficient matrix, thus being

Fig. 6 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.

capable of recovering dictionary atoms under the condition of unfixed and unknown sparsity. In view of this point, NNSC was superior to $\text{NMF}\ell^0\text{-H}$ but performed worse than SHALS. As with the above experiment, SHALS performed best regardless of SNR and sparsity. It is worthy of notice that SHALS performed well not only under conditions of exactly fixed and assigned sparsity but also under conditions of unfixed and unknown sparsity. $\text{NMF}\ell^0\text{-H}$ and NN-KSVD cannot perform like this because they need assigned (exact) sparsities as parameters. Although NMFSC does not depend on an assigned (exact) sparsity, it cannot recover any atoms in the experiment. In view of this point, SHALS has a remarkable advantage, since sparsity is generally unknown and unfixed in practice.

Table 1 also shows the runtime of each trial (see the rightmost column of the table). Obviously, SHALS achieved learning much faster than the other algorithms, and

its runtime did not increase with decreasing sparsity of coefficients. By comparison, NN-KSVD and $\text{NMF}\ell^0\text{-H}$ are just the contrary; their runtime increased with decreasing sparsity of the coefficient matrices.

4.2 Image Denoising of Nature Images

The image denoising problem is important, not only because of the obvious applications that it serves. Being the simplest possible inverse problem, it also 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 the image denoising problem.

Using redundant representations and sparsity as driving forces for denoising of signals constitutes significant progress [27]. 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 the white Gaussian noise with a fixed standard deviation σ , and $\mathbf{Y} \in \mathbb{R}^{m \times n}$ is the noisy observed image. Here, we consider the mode with uniformly distributed positive noise instead of zero-mean white and homogeneous Gaussian noise, in view of the limitations of nonnegative signals studied in this paper. For solving the denoising problem, we adopted the method presented in [27], which is based on a sparse and redundant representation model on small image patches. In the method, image denoising can be solved by minimizing the following optimization problem:

$$\begin{aligned} \{\hat{\alpha}, \hat{\mathbf{D}}, \hat{\mathbf{X}}\} = \arg \min_{\hat{\alpha}, \hat{\mathbf{D}}, \hat{\mathbf{X}}} & \left\{ \lambda \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{j=1}^J \mu_j \|\alpha_j\|_0 \right. \\ & \left. + \sum_{j=1}^J \mu_j \|\mathbf{D}\alpha_j - \mathbf{M}_j \mathbf{X}\|_2^2 \right\} \end{aligned} \quad (6)$$

where λ and μ_j are positive, and the columns of $\alpha \in \mathbb{R}^{K \times J}$ are the coefficient vectors α_j ($j = 1, \dots, J$), in a way that the j -th patch $\mathbf{M}_j \mathbf{X}$ is approximated by $\mathbf{D}\alpha_j$. The vector $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$ is an estimation of the true image. The first term in (6) enforces the matching to the data. The second and the last terms provide regularization, considering that the solution has a sparse representation for every overlapping patch over the learned dictionary $\hat{\mathbf{D}}$. In the procedure, the original dictionary learning method is replaced with our proposed SHALS. For the sparse coding method, the OMP is replaced with NNBP to guarantee the nonnegativity.

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 positive noise of various levels to the test images. To distinguish the positive noise from the Gaussian noise, we used γ to denote the positive noise level. By computing, we found the contrast relationships between σ and

Table 3 PSNR (dB) and SSIM results for different methods. In each cell, five groups of denoising results corresponding to five methods are shown. Top row: ProbShrink-SP [28]; Second row: BLS-GSM [29]; Third row: NL-means [30]; Fourth row: NMF ℓ^0 -H; Bottom row: SHALS.

γ / PSNR	Lena		Barbara		Boat		House		Pepper		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
13.95/29.97	39.10	0.9475	38.46	0.9630	37.61	0.9407	39.07	0.9450	38.55	0.9517	38.56	0.9496
	39.26	0.9515	38.40	0.9667	38.07	0.9503	39.69	0.9558	38.67	0.9615	38.82	0.9572
	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
	39.33	0.9485	38.77	0.9651	37.84	0.9409	40.27	0.9592	38.60	0.9572	38.92	0.9542
43.5/20.12	34.11	0.8922	32.23	0.9094	32.21	0.8579	33.66	0.8697	32.42	0.8951	32.93	0.8849
	34.30	0.8937	31.89	0.9051	32.50	0.8591	34.53	0.8826	33.02	0.9117	33.25	0.8904
	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
	34.45	0.8797	32.89	0.8913	32.52	0.8409	34.76	0.8754	32.97	0.8861	33.52	0.8747
87/14.09	30.77	0.8332	28.22	0.8242	28.88	0.7696	30.54	0.8203	29.01	0.8377	29.48	0.8170
	31.28	0.8432	27.85	0.8155	29.32	0.7792	31.63	0.8408	29.50	0.8530	29.92	0.8263
	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
	31.37	0.7962	29.47	0.7875	29.28	0.7273	31.77	0.8051	29.64	0.7988	30.31	0.7830
160/8.82	27.93	0.7713	24.69	0.6969	26.12	0.6741	27.30	0.7453	25.74	0.7333	26.36	0.7242
	28.64	0.7797	24.77	0.6991	26.64	0.6872	28.87	0.7804	26.54	0.7768	27.09	0.7446
	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
	28.27	0.6692	25.71	0.6139	26.41	0.5821	28.37	0.6888	26.39	0.6664	27.03	0.6441

Table 2 Contrast relationships of σ and γ under the approximately same PSNR conditions.

Parameter	9 dB	14 dB	20 dB	30 dB	42 dB
σ	90	50	25	8	2
γ	160	87	43.5	13.95	5.77

γ under the condition of the approximately same PSNR values as shown in Table 2. We used two quality measures, the peak SNR (PSNR) and the structural similarity (SSIM) [31], to assess the denoised images. Let $\mathbf{X}$ and $\hat{\mathbf{X}}$ denote the ideal image and the deteriorated image, respectively. We calculate the PSNR value of $\hat{\mathbf{X}}$ by $\text{PSNR}(\hat{\mathbf{X}}) = 10 \cdot \log_{10}(1/(\mathbf{X} - \hat{\mathbf{X}})^2)$. For SSIM, its value range is between 0 and 1, and its value equals 1 if $\mathbf{X} = \hat{\mathbf{X}}$.

In the experiment, we focused on tests with higher noise levels, because it may be more critical. We chose the ProbShrink-SP method[†] (using a redundant wavelet representation) [28], the BLS-GSM method^{††} of Portilla *et al.* [29] and the non-local means (NL-means) method^{†††} [30] as the compared objects. These methods are considered to be up-to-date of the efficient and professional denoising methods. Additionally, we also chose the NMF ℓ^0 -H because of its better performance in previous experiments. It is notable that NMF ℓ^0 -H is very time-consuming during dictionary learning, as described in the two experiments above. Table 3 summarizes the results of the denoising experiment.

We concluded that the denoising method using the SHALS dictionary achieved highly competitive PSNR and SSIM performance outcomes compared to those of ProbShrink-SP, BLS-GSM, NL-means, and NMF ℓ^0 -H methods. When comparing PSNR, the denoising method us-

Fig. 7 The SHALS dictionary learned from the noisy “House” image in Fig. 6. The dictionary has a size of 64×256, in which each patch represents an atom of size 8×8.

ing the SHALS dictionary performed much better than NL-means and NMF ℓ^0 -H methods and outperformed NL-means in the range of about 0.7 dB ~ 2 dB. Our approach achieved quite competitive performances compared to the powerful ProbShrink-SP and BLS-GSM, although they are professional denoising methods whereas the proposed method is for dictionary learning with an application for denoising. For making the comparison more understandable, we also showed the results about ProbShrink-SP, BLS-GSM and NL-means in Table 3 as Fig. 8. These results show that, in most regions, the proposed method performs better than the other methods, although the differences are not so big. When comparing the SSIM index, the denoising method using the SHALS dictionary returned results comparable to

[†] Available at: <http://telin.ugent.be/~sanja/>^{††} Available at: <http://decsai.ugr.es/~javier/>^{†††} Available at: <http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter/>

Fig. 8 PSNR comparisons of denoising performance of several efficient methods. Curves depict output PSNR as a function of input PSNR. All results are taken from Table 3. Triangle symbols indicate our results. These results show that, in most regions, the proposed method performs better than the other methods, although the differences are not so big.

those of ProbShrink-SP and BLS-GSM methods at most compared noise levels. A subjective quality comparison for the typical test image “House” is shown in Fig. 6. The SHALS dictionary learned from the noisy “House” image in Fig. 6 is illustrated in Fig. 7.

4.3 Human Face Image Analysis

In this subsection, we describe our experiment on learning a global-based representation [17] 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. 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 performance of the SHALS and the other 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 $S(\mathbf{H})$). By using the Hoyer’s sparsity measure, defined as

$$\text{Sparsity}(\mathbf{x}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1 / \|\mathbf{x}\|_2}{\sqrt{n} - 1} \in [0, 1] \quad (7)$$

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

$$\text{Relative Error} = \frac{\|\mathbf{Y} - \mathbf{WH}\|_F}{\|\mathbf{Y}\|_F} \quad (8)$$

In the experiment, we performed a global-based feature learning of rank $r = 64$ and constrained the coefficient matrices to have a sparsity of about 0.05; i.e., each facial image was required to be represented with three facial features ($64 \times 0.05 \approx 3$). Besides NNSC, NMFSC, and NMF^{ℓ^0} -H, we chose another sparse NMF method (denoted as SNMF) [16] as the compared objective. Note that NN-KSVD was not included in this experiment, since it has exceedingly high computational cost. 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.

Fig. 9 Globally featured faces learned by SNMF, NNSC, NMFSC, $\text{NMF}\ell^0\text{-H}$, and SHALS.

Table 4 Comparisons of $S(\mathbf{H})$ -based sparsity, Hoyer's sparsity (based on (7)), relative error (based on (8)), and runtime for SNMF, NNSC, NMFSC, $\text{NMF}\ell^0\text{-H}$, and SHALS.

Method	$S(\mathbf{H})$	Sparsity(Hoyer)	Relative Error	Time(s)
SNMF	83.01	0.5134	0.9903	1236
NNSC	5.00	0.9925	0.4396	1075
NMFSC	5.23	0.9659	0.1719	1822
$\text{NMF}\ell^0\text{-H}$	4.69	0.9283	0.1641	2871
SHALS	5.00	0.9610	0.2561	65

For NNSC, we also allowed at most 3000 iterations; and for the parameter λ of penalty term and the stepsize μ , we set to 20 and 0.1, respectively. For NMFSC, we only constrained the sparsity of coefficient factor $\mathbf{H}$ to 0.9 in terms of (7) and executed at most 3000 iterations, which was necessary for convergence. For $\text{NMF}\ell^0\text{-H}$, we set the maximum number of nonzero elements of vectors in factor $\mathbf{H}$ to 3 ($3/64 \approx 0.0469$, close to 0.05) and allowed 30 iterations, considering the high computational cost of $\text{NMF}\ell^0\text{-H}$. For the proposed SHALS, we allowed at most 200 iterations, and λ was set to 10 that is calibrated off-line through several trials in order to reduce computational cost. All five algorithms were run three times with the same initial random matrices (for $\text{NMF}\ell^0\text{-H}$, it was not necessary to initialize coefficient $\mathbf{H}$). The averaged results are reported in Table 4.

Through Table 4, 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 four algorithms obtained similar results and produced much sparser solutions; i.e., more global-based representations. NNSC achieved a sparser representation (based on Hoyer's sparsity measure) but suffered from a higher relative error. NMFSC and $\text{NMF}\ell^0\text{-H}$ produced lower relative errors but took extremely more runtime than SHALS. The runtime of NMFSC and $\text{NMF}\ell^0\text{-H}$ were about 28 and 44 times longer than that of SHALS. In view of its high efficiency, SHALS is more suitable for large-scale data analysis. In Fig. 9, we present an illustration of the global-based features learned by the five algorithms in a typical run.

5. Conclusion

In this paper, we presented an NMF-based method for learning overcomplete, nonnegative dictionaries and for obtaining the corresponding coefficients for sparse representation of nonnegative signals. In this method, we forced the coefficients to be as sparse as possible and simultaneously forced the atoms of the dictionary to be as incoherent as possible for obtaining sparser representations. The method was implemented by the proposed sparse hierarchical alternating least squares (i.e., SHALS) algorithm that updates the dictionary in a column-wise and the coefficient matrix in a row-wise manner, which is meant by the terms "alternating" and "hierarchical". In each column-wise or row-wise updating, SHALS optimized a series of optimal problems sequentially, each of which was an optimization of a quadratic function that could be solved explicitly, and thus the whole problem could be efficiently solved.

In addition, we demonstrated that the SHALS algorithm could efficiently solve nonnegative dictionary learning problem. Results of experiments on dictionary recovery showed that SHALS could learn a nonnegative, overcomplete dictionary correctly, regardless of whether the objective signals are synthetic data or natural images. Additionally, further experiments supported the potential application of SHALS in the field of image processing such as image denoising, image classification, and large-scale data processing due to its low computational cost. We are currently working on improving the adjustment to the regularization parameters and on applying this method to some practical problems in image processing; e.g., large-scale image classification. The results from these ongoing studies will be presented in the future.

Acknowledgments

The authors would like to thank the associate editor Professor Kazushi Ikeda and anonymous reviewers for their valuable comments and helpful suggestions that greatly improves this paper's quality.

References

- [1] A. Cichocki and A.H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Trans. Fundamentals*, vol.E92-A, no.3, pp.708–721, March 2009.
- [2] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol.45, no.3, pp.765–777, March 2007.
- [3] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 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 Trans. Inf. Forensics Security*, vol.2, no.3, pp.588–595, Sept. 2007.
- [5] M. Wang, W. Xu, and A. Tang, "A unique "nonnegative" solution to an underdetermined system: From vectors to matrices," *IEEE Trans. Signal Process.*, vol.59, no.3, pp.1007–1016, 2011.
- [6] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol.98, no.6, pp.972–982, 2010.
- [7] D.L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proc. National Academy of Sciences of the United States of America*, vol.100, no.5, pp.2197–2202, March 2003.
- [8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol.32, no.2, pp.407–499, 2004.
- [9] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol.54, no.11, pp.4311–4322, Nov. 2006.
- [10] K. Engan, K. Skretting, and J.H. Husoy, "Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digit. Signal Process.*, vol.17, no.1, pp.32–49, Jan. 2007.
- [11] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol.58, no.4, pp.2121–2130, April 2010.
- [12] M. Aharon, M. Elad, and A.M. Bruckstein, "K-SVD and its non-negative variant for dictionary design," *Proc. SPIE Conference Wavelets*, pp.327–339, 2005.
- [13] D.D. Lee and H.S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol.401, pp.788–791, 1999.
- [14] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, pp.556–562, 2001.
- [15] P.O. Hoyer, "Non-negative sparse coding," In *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pp.557–565, 2002.
- [16] W. Liu, N. Zheng, and X. Lu, "Non-negative matrix factorization for visual coding," 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), pp.293–296, April 2003.
- [17] P.O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Machine Learning Research*, vol.5, pp.1457–1469, 2004.
- [18] V.P. Pauca, F. Shahnaz, M.W. Berry, and R.J. Plemmons, "Text mining using non-negative matrix factorizations," *Proc. 4th SIAM International Conference on Data Mining*, pp.452–456, 2004.
- [19] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol.21, no.21, pp.3970–3975, 2005.
- [20] R. Peharz, M. Stark, and F. Pernkopf, "Sparse nonnegative matrix factorization using ℓ^0 -constraints," 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp.83–88, Sept. 2010.
- [21] R. Peharz and F. Pernkopf, "Sparse nonnegative matrix factorization using ℓ^0 -constraints," *Neurocomputing*, vol.80, pp.38–46, March 2012.
- [22] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol.43, no.1, pp.129–159, 2001.
- [23] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proc. 27th Annu. Asilomar Conf. Signals, Systems and Computers*, pp.40–44, Nov. 1993.
- [24] A.M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations," *IEEE Trans. Inf. Theory*, vol.54, no.11, pp.4813–4820, Nov. 2008.
- [25] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," *IEEE Trans. Signal Process.*, vol.57, no.12, pp.4800–4810, 2009.
- [26] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss-Seidel method under convex constraints," *Oper. Res. Lett.*, pp.127–136, 2000.
- [27] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol.15, no.12, pp.3736–3745, Dec. 2006.
- [28] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multi-band image denoising," *IEEE Trans. Image Process.*, vol.15, no.3, pp.654–665, 2006.
- [29] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol.12, no.11, pp.1338–1351, 2003.
- [30] A. Buades, B. Coll, and J.M. Morel, "A non-local algorithm for image denoising," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.60–65, June 2005.
- [31] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol.13, no.4, pp.600–612, April 2004.

Appendix

We here derive the update rules for $\mathbf{H}$ and $\mathbf{W}$ of (3). In terms of the definition and properties of the Frobenius norm, for a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\|\mathbf{A}\|_F^2 = \text{Tr}(\mathbf{A}\mathbf{A}^T) = \text{Tr}(\mathbf{A}^T\mathbf{A})$. $\text{Tr}(\cdot)$ denotes the trace of a square matrix. Thus, the objective function (3) can be decomposed as follows:

$$J = \sum_{j=1}^n \mathbf{Y}_j^T \mathbf{Y}_{:,j} - 2 \sum_{j=1}^n [\mathbf{Y}^T \mathbf{W}]_{j,:} \mathbf{H}_{:,j} + \sum_{j=1}^n \mathbf{H}_j^T \mathbf{W}^T \mathbf{W} \mathbf{H}_{:,j} + \alpha \sum_{i=1}^r \sum_{j=1}^n |\mathbf{H}_{ij}| + \beta \sum_{i=1}^r \sum_{j=1, j \neq i}^n \mathbf{W}_{i,:}^T \mathbf{W}_{j,:} \quad (\text{A} \cdot 1)$$

where $[\mathbf{Y}^T \mathbf{W}]_j$ denotes the j -th 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 (A·1) can be omitted. If we fix $\mathbf{W}$ in (A·1), then (A·1) is a multivariable objective function of $\mathbf{H}_{ij}$. For (A·1), 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,

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

$$- [\mathbf{W}^T \mathbf{Y}]_{kj} + \frac{\alpha}{2} \Big) \quad (\text{A} \cdot 2)$$

where $[\mathbf{W}^T \mathbf{W}]_{kk}$ denotes the entry in the k -th row and the k -th 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). Here, we set $\lambda = \frac{\alpha}{2}$. Thus, when $\mathbf{H}_{kj} = \frac{[\mathbf{W}^T \mathbf{Y}]_{kj} - \sum_{l=1, l \neq k}^r [\mathbf{W}^T \mathbf{W}]_{kl} \mathbf{H}_{lj} - \lambda}{[\mathbf{W}^T \mathbf{W}]_{kk}}$, $J_{\mathbf{H}_{kj}}$ reaches the minimum. Considering the nonnegativity of factor $\mathbf{H}$, $\mathbf{H}_{kj}$ is set to 0 when $\mathbf{H}_{kj}$ is negative. Note that, when updating $\mathbf{H}_{kj}$, the process involves only the elements $\mathbf{H}_{lj, l \neq k}$ of the j -th 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 one whole row of $\mathbf{H}$ at the same time. Thus, the update rule for $\mathbf{H}$ of (A·1) is given as follows:

$$\begin{aligned} \mathbf{H}_{k:}^* &= \arg \min_{\mathbf{H}_{k:} \geq 0} \|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_F^2 \\ &= \max \left(0, \frac{\mathbf{W}_{k:}^T \mathbf{Y} - \sum_{l=1, l \neq k}^r \mathbf{W}_{k:}^T \mathbf{W}_{l:} \mathbf{H}_{l:} - \lambda}{\mathbf{W}_{k:}^T \mathbf{W}_{k:}} \right) \\ &= \max \left(0, \frac{\mathbf{W}_{k:}^T \mathbf{R}_k - \lambda}{\|\mathbf{W}_{k:}\|_2^2} \right) \end{aligned} \quad (\text{A} \cdot 3)$$

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 (A·1), then (A·1) is a multivariable objective function of $\mathbf{W}_{ij}$. For (A·1), 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 (A·1) and obtain a quadratic function with regard to $\mathbf{W}_{ik}$ as follows:

$$\begin{aligned} J_{\mathbf{W}_{ik}} &= [\mathbf{H}\mathbf{H}^T]_{kk} \mathbf{W}_{ik}^2 + 2\mathbf{W}_{ik} \left(\sum_{l=1, l \neq k}^r \mathbf{W}_{il} [\mathbf{H}\mathbf{H}^T]_{lk} \right. \\ &\quad \left. - [\mathbf{Y}\mathbf{H}^T]_{ik} + \frac{\beta}{2} \sum_{l=1, l \neq k}^r \mathbf{W}_{li}^T \right) \end{aligned} \quad (\text{A} \cdot 4)$$

One can find that (A·4) is very similar to (A·2). Here, let $\mu = \frac{\beta}{2}$. In terms of the properties of a single variable quadratic problem, $J_{\mathbf{W}_{ik}}$ obtains the minimum when $\mathbf{W}_{ik} = \frac{[\mathbf{Y}\mathbf{H}^T]_{ik} - \sum_{l=1, l \neq k}^r \mathbf{W}_{il} [\mathbf{H}\mathbf{H}^T]_{lk} - \mu \sum_{l=1, l \neq k}^r \mathbf{W}_{li}^T}{[\mathbf{H}\mathbf{H}^T]_{kk}}$. Considering the nonnegativity of factor $\mathbf{W}$, $\mathbf{W}_{ik}$ is set to 0 when $\mathbf{W}_{ik}$ is negative. Similar to the update rule for $\mathbf{H}$, $\mathbf{W}$ in (A·1) can update by column. Thus, the update rule for $\mathbf{W}$ of (A·1) is expressed as follows:

$$\begin{aligned} \mathbf{W}_{:k}^* &= \arg \min_{\mathbf{W}_{:k} \geq 0} \|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_F^2 \\ &= \max \left(0, \frac{\mathbf{Y}\mathbf{H}_{:k}^T - \sum_{l=1, l \neq k}^r \mathbf{W}_{:l} \mathbf{H}_{l:} \mathbf{H}_{:k}^T - \mu \sum_{l=1, l \neq k}^r \mathbf{W}_{l:}^T}{\mathbf{H}_{:k} \mathbf{H}_{:k}^T} \right) \\ &= \max \left(0, \frac{\mathbf{R}_k \mathbf{H}_{:k}^T - \mu \sum_{l=1, l \neq k}^r \mathbf{W}_{l:}^T}{\|\mathbf{H}_{:k}\|_2^2} \right) \end{aligned} \quad (\text{A} \cdot 5)$$

Zunyi Tang received the M.S. degree in computer and information systems from the University of Aizu, Japan, in 2009. He is currently pursuing the Ph.D. degree in the Department of Computer and Information Systems, the University of Aizu, Japan. His research interests include sparse representation of signals, blind signal processing, image processing and machine learning.

Shuxue Ding received the M.Sc. degree in physics from Dalian University of Technology, P. R. China, in 1988, and the Ph.D. degree in physics from Tokyo Institute of Technology, Japan, in 1996. From 1989 to 1991 and from 1991 to 1992, respectively, he was an Assistant Professor and Associate Professor in the School of Physics and Optoelectronic Energy at Dalian University of Technology. From 1996 to 1998, he was with Fujisoft-ABC Inc., Japan, where he was involved in algorithm design for telecommunication systems. From 1998 to 2003, he was with Clarion Co., Ltd, Japan, where he engaged in researches in communication and signal processing, especially in speech recognition. From 2003 to 2005, he was a visiting faculty at the University of Aizu, Japan. From 2005 to 2010, he was an Associate Professor in the School of Computer Science and Engineering at the University of Aizu. He is currently a Professor in the School of Computer Science and Engineering, the University of Aizu. He is a member of IEEE, ASA, ACM, and IEICE. Dr. Ding has been engaged in research in a wide range of areas of mathematical and physical engineering, such as statistical signal processing, optimization, neural computation, bioelectromagnetism, and information sciences. In particular, he has devoted himself to blind source separation and independent component analysis, and their applications in acoustic signals and vital signs. Recently, he is also conducting research in brain-style information processing, pattern recognition, compressive sensing and sparse representation. He is also interested in speech and image processing, quantum computation and optimization, quantum and information, and the physics of information.