BLOCKWISE COORDINATE DESCENT SCHEMES FOR SPARSE REPRESENTATION

Bao-Di Liu [a], Yu-Xiong Wang [b], Bin Shen [c], Yu-Jin Zhang [d], and Yan-Jiang Wang [a]

[a] Coll. of Information and Control Engineering, China University of Petroleum, Qingdao 266580 China
[b] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA
[c] Dept. of Computer Science, Purdue University, West Lafayette, IN 47907 USA
[d] Dept. of Electronic Engineering, Tsinghua University, Beijing 100084 China
{thu.liubaodi, albertwyx, stanshenbin}@gmail.com
zhang-yj@mail.tsinghua.edu.cn, yjwang@upc.edu.cn

ABSTRACT

The current sparse representation framework is to decouple it as two subproblems, i.e., alternate sparse coding and dictionary learning using different optimizers, treating elements in bases and codes separately. In this paper, we treat elements both in bases and codes homogenously. The original optimization is directly decoupled as several blockwise alternate subproblems rather than above two. Hence, sparse coding and bases learning optimizations are coupled together. And the variables involved in the optimization problems are partitioned into several suitable blocks with convexity preserved, making it possible to perform an exact block coordinate descent. For each separable subproblem, based on the convexity and monotonic property of the parabolic function, a closed-form solution is obtained. Thus the algorithm is simple, efficient and effective. Experimental results show that our algorithm significantly accelerates the learning process.

Index Terms— Dictionary learning, coordinate descent, sparse coding

1. INTRODUCTION

Sparse representation has now been widely used in visual tasks,such as image classification [1, 2, 3], image inpainting [4], image annotation [5, 6, 7]. By coding under over-complete bases, it makes the attained sparse codes capable of representing data more adaptively. Dictionary learning thus involves accordingly, which tries to build dictionary to find atoms identifying the best causes of the target data

At early stages, sparse coding is typical cast as sparse approximation with l_0-norm constraint optimization, which is solved by methods such as Matching Pursuit [8] and Orthogonal Matching Pursuit [9]. Later, its convex relaxation with l_1-norm is widely accepted [10]. Hence, the solution to sparse coding becomes l_1-regularized least square optimization problem ($L1$-LS). Existing approaches to solving $L1$-LS include active-set methods, such as Homotopy [11], LARS [12], and feature-sign search [13], gradient methods (also called first-order methods or iterative soft-thresholding methods), such as operator-splitting [14], iterative splitting and thresholding [15], and fixed-point iteration [16]. Active-set methods are efficient for small or medium-sized problems, or when requiring very sparse solution. Gradient methods need more iterations especially when the solution is not very sparse or the initialization is not ideal. Meanwhile, dictionary learning can be considered to optimize a least squares problem with quadratic constraints ($L2$-LS). Currently, effective algorithms involve MOD [17], K-SVD [18], gradient descent [19], and Lagrange-Dual [13]. MOD updates all the entries of the bases simultaneously, but it is not warranted to converge. K-SVD sequentially updates the bases column-wise together with the corresponding sparse codes using singular value decomposition (SVD), which needs to consume a lot of computation. Gradient descent often shows slow convergence.

Here we consider simultaneous sparse coding and dictionary learning commonly formulated as alternating $L1$-LS and $L2$-LS optimization problems. Among existing approaches, the feature-sign search/ lagrange Dual (FS–LD) algorithm [13] speeds up the sparse coding procedure significantly and achieves optimum performance up to now. However, the cyclical sign feedback and adjustment actually abases its efficiency in feature-sign search. As for the Lagrange Dual algorithm, the adopted Newton's method needs several iterations with low convergence rate. Besides, the presence of the matrix inversion will introduce numerical difficulties in some situations.

The problem is, instead, recast under a much simpler scheme, i.e., blockwise coordinate descent. Coordinate descent algorithms actually were proposed in solving the sparsity induced least square minimization long ago. However, its powerful performance, i.e., efficiency and effectiveness, haven't been fully appreciated. There are only related work in other domains such as non-negative matrix factorization [20, 21, 22]. We can see how it will be revealed by appropriate partition of the variables and resort to simple update rules. This is the major contribution of this paper. In short, if we try to focus on one basic single variable, a direct closed-form solution will be obtained based on the property of a much simpler univariate parabolic function. The analytical solutions of several variables can be further unified into a parallel vector formula according to the separability of the objective function. In addition, this optimization scheme is suitable for both $L1$-LS and $L2$-LS with only slight modifications. Hence, the proposed algorithm is simple, efficient and effective with theoretically warranted convergence. We demonstrate that our algorithm significantly accelerates the solution to sparse coding and dictionary learning, and has superior solutions especially in the case of relatively small number of samples or seeking for comparatively much sparser codes.

The rest of this paper is organized as follows. In Section 2 problem statement is reviewed briefly. The proposed algorithm in solving $L1$-LS and $L2$-LS as well as convergence analysis is elaborated in Section 3. Section 4 shows experimental results and analysis. Discussions and conclusions are drawn in Section 5.

This work was supported in part by National Natural Science Foundation of P.R. China under Grant No.61171118 and No.61271407

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2. PROBLEM STATEMENT

Let $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ be the input data matrix, where D and N are the dimension and number of the data vectors, respectively. Let $\boldsymbol{B} \in \mathbb{R}^{D\times K}$ and $\boldsymbol{S} \in \mathbb{R}^{K\times N}$ denote the basis matrix and corresponding sparse codes (also called coefficient matrix), respectively, where K is the number of the bases. Sparse representation aims to solve the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{S}} f(\boldsymbol{B},\boldsymbol{S}) \;&=\; \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 \\ s.t. \quad & \|\boldsymbol{B}_{\bullet i}\|_2 \le 1,\ \forall i = 1,2,\ldots,K. \end{aligned} \tag{1}$$

Here, $\boldsymbol{A}_{\bullet n}$ and $\boldsymbol{A}_{k\bullet}$ denote the n-th column and k-th row vectors of matrix $\boldsymbol{A}$, respectively. The l_1-norm regularization term is adopted to enforce sparsity of $\boldsymbol{S}$ and α is the regularization parameter to control the tradeoff between fitting goodness and sparseness. It can be decoupled into the following two optimization subproblems which can be solved by alternating minimizations [13].

$L1$-LS minimization problem:

$$\min_{\boldsymbol{S}} f(\boldsymbol{S}) = \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1. \tag{2}$$

$L2$-LS minimization problem:

$$\min_{\boldsymbol{B}} f(\boldsymbol{B}) = \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 \qquad s.t. \|\boldsymbol{B}_{\bullet i}\|_2 \le 1, \forall i = 1,2,\ldots,K. \tag{3}$$

3. PROPOSED ALGORITHM FOR SPARSE CODING AND DICTIONARY LEARNING

Consider the $L1$-LS minimization problem first. If we concentrate on the basic element, i.e., one single variable, with the remaining ones fixed at a time, the objective functions in (2) reduces to a much simpler univariate parabolic function. Thus, a direct closed-form solution minimizing the corresponding cost function can be easily obtained based on the convexity and monotonic property of the parabolic function. Moreover, according to the separability of the objective function, the analytical solutions of several independent variables (in this case are the entries in the same row) can be unified into higher block formula, making parallel computation and further acceleration possible. In other words, upon such block partition mode, an exact block coordinate descent can be carried out effectively. As for the $L2$-LS minimization problem, this strategy is also applicable. In this sense, the solutions to these two optimization problem can be tackled under the same scheme. The specific block partition modes and corresponding update rules of finding sparse codes and learning bases are validated by the following two theorems, respectively.

3.1. $L1$-LS minimization for finding sparse codes

Theorem 1. *$\forall k \in \{1,2,\ldots,K\}$, with $\{\boldsymbol{S}_{p\bullet,p=1,2,\ldots,K}\}/\boldsymbol{S}_{k\bullet}$ and $\boldsymbol{B}$ fixed, the minimization of (2) with respect to the single row has the closed-form solution*

$$\begin{aligned} \boldsymbol{S}_{k\bullet} \;&=\; \arg\ \min_{\boldsymbol{S}_{k\bullet}} \left\{\|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1\right\} \\ &=\; \max\left\{[\boldsymbol{B}_{\bullet k}]^T\boldsymbol{X} - [\boldsymbol{B}_{\bullet k}]^T\boldsymbol{B}\tilde{\boldsymbol{S}}^k, \alpha\right\} \\ &\quad +\min\left\{[\boldsymbol{B}_{\bullet k}]^T\boldsymbol{X} - [\boldsymbol{B}_{\bullet k}]^T\boldsymbol{B}\tilde{\boldsymbol{S}}^k, -\alpha\right\}, \end{aligned} \tag{4}$$

where $\tilde{\boldsymbol{S}}^k_{p\bullet} = \begin{cases} \boldsymbol{S}_{p\bullet}, & p \ne k \\ \boldsymbol{0}, & p = k \end{cases}$.

Proof. The objective function in (2) can be rewritten as

$$\begin{aligned} f(\boldsymbol{S}) \;&=\; \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1 \\ &=\; tr\left\{\boldsymbol{X}^T\boldsymbol{X} - 2\boldsymbol{X}^T\boldsymbol{B}\boldsymbol{S} + \boldsymbol{S}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{S}\right\} + 2\alpha\|\boldsymbol{S}\|_1 \\ &=\; tr\left\{\boldsymbol{X}^T\boldsymbol{X}\right\} - 2\sum_{n=1}^{N}[\boldsymbol{X}^T\boldsymbol{B}]_{n\bullet}\boldsymbol{S}_{\bullet n} \\ &\quad + \sum_{n=1}^{N}\boldsymbol{S}_{\bullet n}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{S}_{\bullet n} + 2\alpha\sum_{k=1}^{K}\sum_{n=1}^{N}|\boldsymbol{S}_{kn}|, \end{aligned} \tag{5}$$

where $tr\{\boldsymbol{A}\}$ represents the trace of matrix $\boldsymbol{A}$.

Ignoring the constant term $tr\left\{\boldsymbol{X}^T\boldsymbol{X}\right\}$, the objective function of $\boldsymbol{S}_{\bullet n}$ reduces to (6) with $\boldsymbol{B}$ fixed.

$$f(\boldsymbol{S}_{\bullet n}) \;=\; \boldsymbol{S}_{\bullet n}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{S}_{\bullet n} - 2[\boldsymbol{X}^T\boldsymbol{B}]_{n\bullet}\boldsymbol{S}_{\bullet n} + 2\alpha\sum_{k=1}^{K}|\boldsymbol{S}_{kn}|. \tag{6}$$

And then the objective function of $\boldsymbol{S}_{kn}$ in (6) reduces to (7) with $\boldsymbol{B}$ and $\{\boldsymbol{S}_{1n},\boldsymbol{S}_{2n},\ldots,\boldsymbol{S}_{kn}\}/\boldsymbol{S}_{kn}$ fixed.

$$\begin{aligned} f(\boldsymbol{S}_{kn}) \;&=\; \boldsymbol{S}_{kn}^2[\boldsymbol{B}^T\boldsymbol{B}]_{kk} + 2\alpha|\boldsymbol{S}_{kn}| \\ &\quad +2\boldsymbol{S}_{kn}\left\{\sum_{l=1,l\ne k}^{K}[\boldsymbol{B}^T\boldsymbol{B}]_{kl}\boldsymbol{S}_{ln} - [\boldsymbol{B}^T\boldsymbol{X}]_{kn}\right\} \\ &=\; \boldsymbol{S}_{kn}^2[\boldsymbol{B}^T\boldsymbol{B}]_{kk} + 2\alpha|\boldsymbol{S}_{kn}| - 2\boldsymbol{S}_{kn}\boldsymbol{H}_{kn}, \end{aligned} \tag{7}$$

where $\boldsymbol{H}_{kn}=[\boldsymbol{B}^T\boldsymbol{X}]_{kn} - \sum_{l=1,l\ne k}^{K}[\boldsymbol{B}^T\boldsymbol{B}]_{kl}\boldsymbol{S}_{ln}$.

When $\|\boldsymbol{B}_{\bullet k}\|_1 > 0$,$f(\boldsymbol{S}_{kn})$ is piece-wise parabolic function with $[\boldsymbol{B}^T\boldsymbol{B}]_{kk} = 1$. Based on the convexity and monotonic property of the parabolic function, it is not difficult to know that $f(\boldsymbol{S}_{kn})$ reaches the minimum at the unique point.

$$\boldsymbol{S}_{kn} = \max\{\boldsymbol{H}_{kn},\alpha\} + \min\{\boldsymbol{H}_{kn},-\alpha\}. \tag{8}$$

Furthermore,given that the optimal value for $\boldsymbol{S}_{kn}$ does not depend on the other entries in the same row, each whole row of $\boldsymbol{S}$ can be optimized simultaneously. That is

$$\boldsymbol{S}_{k\bullet} = \max\{\boldsymbol{H}_{k\bullet},\alpha\} + \min\{\boldsymbol{H}_{k\bullet},-\alpha\}\ \textbf{Q.E.D} \tag{9}$$

3.2. $L2$-LS minimization for learning dictionary

Theorem 2. *$\forall k \in \{1,2,\ldots,K\}$, with $\boldsymbol{S}$ and $\{\boldsymbol{B}_{\bullet q,q=1,2,\ldots,K}\}/\boldsymbol{B}_{\bullet k}$ fixed, the constrained minimization problem of (3) with respect to the single column has the closed-form solution*

$$\begin{aligned} \boldsymbol{B}_{\bullet k} \;&=\; \arg\ \min_{\boldsymbol{B}_{\bullet k}} \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2 \\ &=\; \frac{\boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T - \tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T}{\left\|\boldsymbol{X}[\boldsymbol{S}_{k\bullet}]^T - \tilde{\boldsymbol{B}}^k\boldsymbol{S}[\boldsymbol{S}_{k\bullet}]^T\right\|_2}, \end{aligned} \tag{10}$$

where $\tilde{\boldsymbol{B}}^k_{\bullet p} = \begin{cases} \boldsymbol{B}_{\bullet p}, & p \ne k \\ \boldsymbol{0}, & p = k \end{cases}$.

Proof. Without the sparseness regularization term in (2) and additional constraints in (3), $\boldsymbol{S}_{k\bullet}$ and $\boldsymbol{B}_{\bullet k}$ are dual in objective function $\|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{S}\|_F^2$ for $\forall k \in \{1,2,\ldots,K\}$. Using similar derivation procedures and additional projection to the feasible region, this can be proven.

3.3. Overall algorithm

Our algorithm for sparse coding and bases learning is shown in Algorithm 1. Here,$\boldsymbol{1} \in \mathbb{R}^{K\times K}$ is a square matrix with all elements 1, $\boldsymbol{I} \in \mathbb{R}^{K\times K}$ is the identity matrix, and $\odot$ indicates element dot product. By iterating $\boldsymbol{S}$ and $\boldsymbol{B}$ alternately, the sparse codes are obtained, and the corresponding bases are learned.

Algorithm 1 Blockwise Coordinate Descent for Dictionary Learning

Require: Data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ and K

$\boldsymbol{B}\leftarrow rand(D,K)$, $\boldsymbol{B}_{\bullet k}=\dfrac{\boldsymbol{B}_{\bullet k}}{\|\boldsymbol{B}_{\bullet k}\|_2}\forall k$, $\boldsymbol{S}\leftarrow zeros(K,N)$

$iter = 0$

while $(f(iter)-f(iter+1))/f(iter) > 1e-6$ **do**

  $iter \leftarrow iter + 1$

  Update $\boldsymbol{S}$:

  Compute $\boldsymbol{A} = (\boldsymbol{B}^T\boldsymbol{B}) \odot (\boldsymbol{1} - \boldsymbol{I})$ and $\boldsymbol{E} = \boldsymbol{B}^T\boldsymbol{X}$

  for $k = 1;k \le K;k{+}{+}$ **do**

    $\boldsymbol{S}_{k\bullet} = \max\{\boldsymbol{E}_{k\bullet} - \boldsymbol{A}_{k\bullet}\boldsymbol{S}, \alpha\}$
    $+ \min\{\boldsymbol{E}_{k\bullet} - \boldsymbol{A}_{k\bullet}\boldsymbol{S}, -\alpha\}$

  end for

  Update $\boldsymbol{B}$:

  Compute $\boldsymbol{G} = (\boldsymbol{S}\boldsymbol{S}^T) \odot (\boldsymbol{1}-\boldsymbol{I})$, $\boldsymbol{W} = \boldsymbol{X}\boldsymbol{S}^T$

  for $k = 1;k \le K;k{+}{+}$ **do**

    $\boldsymbol{B}_{\bullet k} = \dfrac{\boldsymbol{W}_{\bullet k} - \boldsymbol{B}\boldsymbol{G}_{\bullet k}}{\|\boldsymbol{W}_{\bullet k} - \boldsymbol{B}\boldsymbol{G}_{\bullet k}\|_2}$

  end for

  Update the objective function:

  $f(iter) = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + 2\alpha\|\boldsymbol{S}\|_1$

end while

return $\boldsymbol{B}$, and $\boldsymbol{S}$

3.4. Analysis and Comment

Theorem 3. *The objective function in* (1) *is nonincreasing under the update rules given in* (4) *and* (10).

Proof. Since the exact minimization point is obtained by (4) or (10), each operation updates $\boldsymbol{S}_{1\bullet}, \ldots, \boldsymbol{S}_{K\bullet}, \boldsymbol{B}_{\bullet 1}, \ldots, \boldsymbol{B}_{\bullet K}$ alternately, it monotonically decreases the objective function in (1). Considering that the objective function is obviously bounded below, it converges. **Q.E.D.**

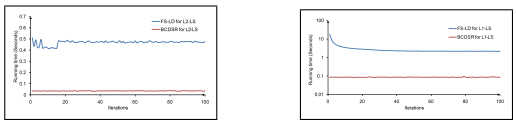
4. EXPERIMENTAL RESULTS

The performance of our blockwise coordinate descent for dictionary learning (BCDDL) algorithm is evaluated on three common datasets: natural images [23], Caltech-101 [24], and Scene 15 [25]. All experiments are implemented on PC with Intel Core i5 M560 2.67GHz CPU, 8GB RAM. The software environment is Windows 7 and Matlab 7.12.

For natural images dataset, patches are randomly selected. In each patch, every pixel value forms a vector to represent this patch. For Caltech-101 and Scene 15, 16×16 size patches are densely sampled from images and are then represented by SIFT descriptor with grid size 4×4. Given that FS–LD algorithm significantly outperforms other approaches [13], a systematic evaluation and comparison mainly between proposed method and FS–LD algorithm are carried out with regard to their performance in terms of running time, convergence rate, learning bases in certain cases, and the like.

4.1. Running time for learning bases and finding sparse codes

In this section, the algorithm is assessed on natural images dataset with a set of 1000 input vectors (each 14×14 pixels) randomly selected. α is set to 0.2. The number of bases is 512. We compare our algorithm with FS–LD algorithm. To avoid being influenced by random variations of environment, the experiment is repeated 20 times. In each experiment, 100 iterations are operated to optimize bases and sparse codes alternately. Figure 1 shows the running time per iteration. Figure 1 (a) shows the running time for $L2$-LS. Figure 1 (b) shows the running time for $L1$-LS. It is obvious that our algorithm runs much faster for both $L2$-LS and $L1$-LS per iteration. The running time of FS–LD algorithm is especially long in the first few iterations, because the energy is scattered in the initial stages which requires more feedback and adjustment to determine the response of the sparse codes. The average ratio of running time per iteration of FS–LD algorithm to that of ours is about 12.98 and 31.86 for $L2$-LS and $L1$-LS, respectively.



(a) $L2$-LS (b) $L1$-LS

Fig. 1. Comparison of running time per iteration between BCDDL and FS–LD algorithm.

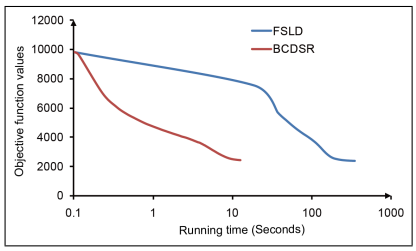


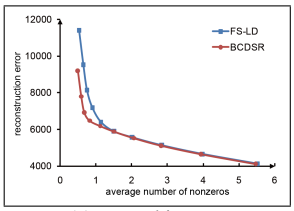
Fig. 2. Comparison of the convergence between BCDDL and FS–LD algorithm.

4.2. Convergence rate

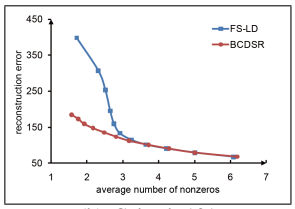
The speed of convergence is another important factor to evaluate algorithms. Here, a set of 1000 input vectors (each 14×14 pixels) randomly sampled from natural images dataset are used. α is set to 0.2, the number of bases is 512, and 100 iterations are operated to learn bases and find sparse codes alternately. Figure 2 shows the convergence rate comparison between these two algorithms, where our algorithm converges much faster. Hence, when combining the experimental results in Figure 1 and 2, it demonstrates that our algorithm has fast convergence with low cost per iteration.

4.3. Total time for dictionary learning

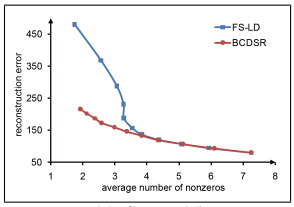
With respect to the running time per iteration, our BCDDL algorithm runs much faster than FS–LD algorithm. We then evaluate the separate stage for learning dictionary on the above three datasets, which is the core concern in training. A set of 1000 input vectors (each 14×14 pixels for natural images and 128-D SIFT for Caltech-101 and Scene 15) are chosen, respectively. For natural images dataset, α is set to 0.2 and the number of bases is 512; for Caltech-101 and Scene 15 datasets, α is set to 0.1 and the number of bases is also 512. The stopping condition is that the relative change of the objective function value between successive iterations is less than $1e-6$ (i.e. $(fold - fnew)/fold < 1e-6$). The running time of our BCDDL algorithm is $71.29s$, $30.82s$, and $25.53s$ in natural images, Caltech-101, and Scene 15 datasets, respectively, while FS–LD algorithm $1452.50s$, $231.48s$, and $398.22s$. Hence, the total time for



(a) natural images

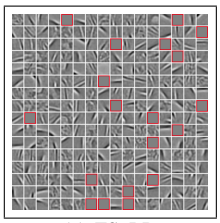


(b) Caltech-101

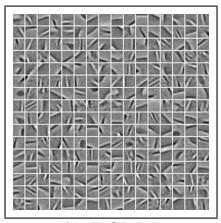


(c) Scene 15

Fig. 3. Comparison of the relationship between reconstruction error and sparsity.



(a) FS–LD



(b) BCDDL

Fig. 4. Comparison of learned dictionary with extremely sparse codes.

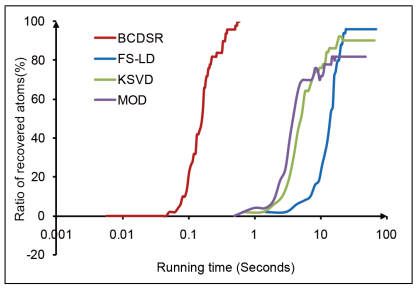


Fig. 5. Comparison of dictionary learning behavior among BCDDL and classical approaches.

dictionary learning also demonstrates that the speed of our BCDDL algorithm outperforms FS–LD algorithm significantly.

4.4. The relationship between reconstruction error and sparsity

The effectiveness of our algorithm can be evaluated by the relationship between reconstruction error and sparsity to some extent. Here the sparsity is defined as the average of number of nonzero entities in each column of sparse codes matrix. Apart from the remarkable speedup presented above, Figure 3 shows such variation tendency. The conditions are similar to the previous experiment conditions, except that for natural images dataset, α is set to $0.6 : 0.1 : 1.5$ and the number of bases is 256, while for Caltech-101 and Scene 15 datasets, α is set to $0.09 : 0.02 : 0.3$ and the number of bases is 256.

From Figure 3, both algorithms achieve the approximately equal reconstruction error when the corresponding codes are not too sparse. However, our algorithm attains less reconstruction error as for lower sparsity values which corresponds to higher sparseness. This indicates that our algorithm is capable of finding comparatively much sparser codes while maintaining lower reconstruction error, which is helpful for some real-world applications demonstrated in the following sections.

4.5. Learning bases with extremely sparse codes cases

Figure 4 gives a comparison of learned natural images bases between these two algorithms when the codes are extremely sparse. Figure 4(a) is the result of FS–LD algorithm, and Figure 4(b) is the result of BCDDL algorithm. $120,000$ input vectors (each 14×14 pixels) are utilized to infer a set of 256 bases in both cases. α is set to 2.0 and 100 iterations are operated for dictionary learning. Notice that there are several basis images with all zero pixel values in Figure 4(a) (The regions marked with the red boxes), which implies that the corresponding basis vectors make no sense. Reversely, our BCDDL algorithm is still adaptive for such situation. So even in the case of extremely sparse codes, our algorithm remains effective.

4.6. Comparison on a synthetic experiment

To demonstrate the BCDDL algorithm, a synthetic experiment is carried out. A random dictionary (iid Gaussian entries, normalized columns) of size 20×50 is generated. From the generated dictionary, $1,500$ samples are produced by a random combination of 3 atoms, with coefficients drawn from the normal distribution $\mathcal{N}(0, 1)$. For each samples, it is contaminated by a random zero-mean Gaussian noise with signal-to-noise ratio of 20dB. 100 iterations is carried out to recover the original dictionary by four common dictionary learning methods. Figure 5 shows the ratio of recovered atoms. From Figure 5, we can see that, our BCDDL algorithm is capable of recovering 100% of atoms in 0.58 second. FS–LD algorithm recovered 96% of atoms in 70.07 second. KSVD algorithm recovered 90% of atoms in 65.64 second. MOD algorithm recovered 82% of atoms in 47.70 second.

5. CONCLUSION

In this paper, we reformulate the optimization problems in a new fashion for sparse coding and dictionary learning. It is probably the fastest procedure for SC-DL to-date. Two highlights distinguish it from previous works.

1 The simplest coordinate descent does work in SC-DL and deserves more attention. Our exhaustive experiments have demonstrated that BCDDL is surprisingly competitive in seeking better solutions much faster.

2. The efficiency of BCDDL not only lies on coordinate descent, but also owes to its proper partition of variables, making parallel computation feasible. This means BCDDL is blockwise rather than coordinate-wise.

6. REFERENCES

[1] Bao-Di Liu, Yu-Xiong Wang, Yu-Jin Zhang, and Yin Zheng, "Discriminant sparse coding for image classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2193–2196.

[2] Bao-Di Liu, Yu-Xiong Wang, Yu-Jin Zhang, and Bin Shen, "Learning dictionary on manifolds for image classification," *Pattern Recognition*, vol. 46, no. 7, pp. 1879–1890, 2013.

[3] Bao-Di Liu, Yu-Xiong Wang, Bin Shen, Yu-Jin Zhang, Yan-Jiang Wang, and Wei-Feng Liu, "Self-explanatory convex sparse representation for image classification," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2120–2125.

[4] Bin Shen, Wei Hu, Yimin Zhang, and Yu-Jin Zhang, "Image inpainting via sparse representation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 697–700.

[5] Weifeng Liu, Dacheng Tao, Jun Cheng, and Yuanyan Tang, "Multiview hessian discriminative sparse coding for image annotation," *Computer Vision and Image Understanding*, vol. 118, pp. 50–60, 2014.

[6] Dacheng Tao and Weifeng Liu, "Multiview hessian regularization for image annotation," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2676–2687, 2013.

[7] W Liu, H Liu, D Tao, Y Wang, and K Lu, "Manifold regularized kernel logistic regression for web image annotation," *arXiv preprint arXiv:1312.6180*, 2013.

[8] S.G. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[9] Y.C. Pati, R. Rezaiifar, and PS Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," 1993, pp. 40–44.

[10] David L. Donoho, "For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution," *Journal of Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 907–934, 2006.

[11] M.R. Osborne, B. Presnell, and B.A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389, 2000.

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

[13] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, "Efficient sparse coding algorithms," 2007, vol. 19, pp. 801–808.

[14] P.L. Combettes and V.R. Wajs, "Signal recovery by proximal forward-backward splitting," *Journal of Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2006.

[15] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Journal of Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[16] E.T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for l1-minimization: Methodolgy and convergence," *SIAM Jouranl on Optimization*, vol. 19, 2008.

[17] K. Engan, S.O. Aase, and J.Hakon Husoy, "Method of optimal directions for frame design," in *In proceedings of 24th ICASSP*, 1999, vol. 5, pp. 2443–2446.

[18] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[19] Y. Censor and S.A. Zenios, *Parallel optimization: Theory, algorithms, and applications*, Oxford University Press, USA, 1997.

[20] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari, "Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization," in *Independent Component Analysis and Signal Separation*, pp. 169–176. Springer, 2007.

[21] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie, "Fast nonnegative matrix/tensor factorization based on low-rank approximation," *Signal Processing, IEEE Transactions on*, vol. 60, no. 6, pp. 2928–2940, 2012.

[22] Y Wu, B Shen, and H Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2014.

[23] Bruno. A. Olshausen and David. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[24] Fei-Fei Li, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," 2004, vol. 12, p. 178.

[25] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," 2006.