SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Semi-Blind Compressed Sensing via Adaptive Dictionary Learning and One-Pass Online Extension

Di Ma[1,2] & Songcan Chen[1,2*]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106*, China;*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing* 211106*, China*

Appendix A The Algorithms

Appendix A.1 Algorithm A1

Algorithm A1 FISTA

Input: recovered data $\mathbf{x}$, learned dictionary $\mathbf{D}$, regularization parameters λ, α, Lipschitz constant L of ∇f ($f(\mathbf{s}) \coloneqq \frac{\lambda}{2}\|\mathbf{x} - \mathbf{D}\mathbf{s}\|_2^2$ in our algorithm)
Output: $\mathbf{s}$

initialize $\mathbf{z}^1 = \mathbf{s}^0 = \mathbf{D}^T\mathbf{x}$, $t^1 = 1$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{s}^k = \arg\min_{\mathbf{s}} \left\{ \alpha\|\mathbf{s}\|_1 + \frac{L}{2} \left\| \mathbf{s} - (\mathbf{z}^k - \frac{1}{L}\nabla f(\mathbf{z}^k)) \right\|_2^2 \right\}$
  $t^{k+1} = \frac{1 + \sqrt{1 + 4(t^k)^2}}{2}$
  $\mathbf{z}^{k+1} = \mathbf{s}^k + \frac{t^k - 1}{t^{k+1}} \left(\mathbf{s}^k - \mathbf{s}^{k-1}\right)$
end for

Appendix A.2 Algorithm A2

Algorithm A2 S-BCS

Input: prior sparsity basis $\mathbf{D}_0$, observed measurements $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$, measurement matrices $\mathbf{\Phi}_i, i = 1, \ldots, n$, hyper-parameter α, β, λ
Output: task-dependent dictionary $\mathbf{D}_T$, final recovery of the data $\mathbf{X}_T$

Initialize $\mathbf{S}_0$, $t = 0$
while not converged **do**
  Learn the intermediate recovery $\mathbf{X}_t$ and the sparse representation $\mathbf{S}_t$ by solving optimization problem (5)
  Learn the gradual transition $\Delta\mathbf{D}_t$ using equation (8)
  Update the dictionary $\mathbf{D}_{t+1}$ using equation (9)
  $t = t + 1$
end while

* Corresponding author (email: s.chen@nuaa.edu.cn)

Appendix B Extension of the Proposed Method

In the previous section, we give our algorithm for adaptively transferring the prior sparsity basis to the task-dependent one when all the compressive measurements have been acquired. Benefiting from the prior knowledge and the incremental learning strategy, our method can be adapted to address online tasks with mini-batch such that the memory cost for large scale problem can be further reduced. In this section, we extend our method to an one-pass online version to adaptively learn a task-dependent dictionary for the above mentioned data types, termed as OS-BCS. Without loss of generality, we formulate the online version with data coming one by one.

The optimization problems about the intermediate recovery $\mathbf{X}_t$ and the sparse representation $\mathbf{S}_t$ can be easily decoupled for each data point. At the arrival of the t-th compressive measurements, let $\mathbf{D}_t = \mathbf{D}_{t-1}$ where $\mathbf{D}_{t-1}$ denotes the learned dictionary before the t-th sample arrival. The optimization problems of $\mathbf{x}_t$ and $\mathbf{s}_t$ can be written as

$$\{\mathbf{x}_t, \mathbf{s}_t\} = \arg\min_{\mathbf{x},\mathbf{s}} \frac{1}{2}\|\mathbf{y}_t - \mathbf{\Phi}_t\mathbf{x}\|_2^2 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{D}_t\mathbf{s}\|_2^2 + \alpha\|\mathbf{s}\|_1 \tag{B1}$$

The update of the next dictionary is different from that of $\mathbf{x}_t$ and $\mathbf{s}_t$. Specifically, after the arrival of the t-th compressive measurements, we use the previously learned intermediate recovery $\{\mathbf{x}_i\}_{i=1}^t$ and the sparse representation $\{\mathbf{s}_i\}_{i=1}^t$ to establish a surrogate cost function and minimize it to obtain a gradual transition of the dictionary. The corresponding optimization problem is as follows

$$\Delta\mathbf{D}_t = \arg\min_{\Delta\mathbf{D}} \frac{1}{t}\sum_{i=1}^{t} \|\mathbf{x}_i - \mathbf{D}_t\mathbf{s}_i - \Delta\mathbf{D}\mathbf{s}_i\|_2^2 + \beta\|\Delta\mathbf{D}\|_F^2 \tag{B2}$$

Then the next dictionary can be updated by using

$$\mathbf{D}_t = \mathbf{D}_t + \Delta\mathbf{D}_t \tag{B3}$$

The whole process of the one-pass online version is summarized in Algorithm B1. Here, to improve the convergence, we update $\Delta\mathbf{D}$ with a warm restart using Algorithm B2 [1].

Algorithm B1 OS-BCS

Input: prior sparsity basis $\mathbf{D}_0$, a sequence of observed measurements $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, measurement matrices $\mathbf{\Phi}_i, i = 1, \ldots, T$, hyperparameter α, β, λ
Output: task-dependent dictionary $\mathbf{D}_T$, final recovery of the data $\mathbf{X}_T$

for $t = 1$ to T **do**
  Initialize $\mathbf{s}_t$, $\mathbf{D}_t = \mathbf{D}_{t-1}$
  while not converged **do**
    Learn the intermediate recovery $\mathbf{x}_t$ and the sparse representation $\mathbf{s}_t$ by solving optimization problem (B1)
    $\mathbf{A}_t \leftarrow \frac{t-1}{t}\mathbf{A}_{t-1} + \frac{1}{t}\mathbf{s}_t\mathbf{s}_t^T$, $\mathbf{B}_t \leftarrow \frac{t-1}{t}\mathbf{B}_{t-1} + \frac{1}{t}\mathbf{x}_t\mathbf{s}_t^T$
    Learn the gradual transition $\Delta\mathbf{D}_t$ using Algorithm B2
    Update the dictionary $\mathbf{D}_t$ using equation (B3)
  end while
end for

Algorithm B2 The Basis Update

Input: $\mathbf{D}$, $\Delta\mathbf{D} = [\Delta\mathbf{d}_1, \ldots, \Delta\mathbf{d}_r] \in \mathbb{R}^{d\times r}$, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_r] \in \mathbb{R}^{r\times r}$, $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{d\times r}$
Output: $\Delta\mathbf{D}$

$\tilde{\mathbf{A}} = \mathbf{A} + \beta\mathbf{I}$, $\tilde{\mathbf{B}} = \mathbf{B} - \mathbf{D}\mathbf{A}$
for $j = 1$ to r **do**
  $\Delta\mathbf{d}_j \leftarrow \frac{1}{\tilde{\mathbf{a}}_{jj}}(\tilde{\mathbf{b}}_j - \Delta\mathbf{D}\tilde{\mathbf{a}}_j) + \Delta\mathbf{d}_j$
end for

Appendix C Experiments

In this section, we conduct experiments on six real-world datasets, including two hyperspectral datasets, Cuprite and Jasper [2], and four publicly available datasets, SensIT [3], USPS [4], madelon [5] and isolet [6], to demonstrate the effectiveness of our proposed method. The description of the datasets used in our experiments are summarized in Table C1.

We compare our method with several conventional algorithms of CS and BCS, including FISTA [7], Laplace [8] and CBK-SVD [9]. For the algorithms based on Bayesian framework, no parameter is tuned. For the rest algorithms, FISTA and our method have parameters to control the sparse regularization, thus we tune them and present the results corresponding to the optimal parameter. Concretely, the parameter α for FISTA and our method are both tuned from $\{10^{-3}, 5\times 10^{-3}, 10^{-2}, \ldots, 10^{2}, 5\times 10^{2}, 10^{3}\}$ via limited validation on a validation set. In our experiment, we fix the size of the validation set as $n_{val} = 100$ and vary the size of the testing set n_{tst} according to different settings of the experiments. The

Table C1 Details of the datasets.

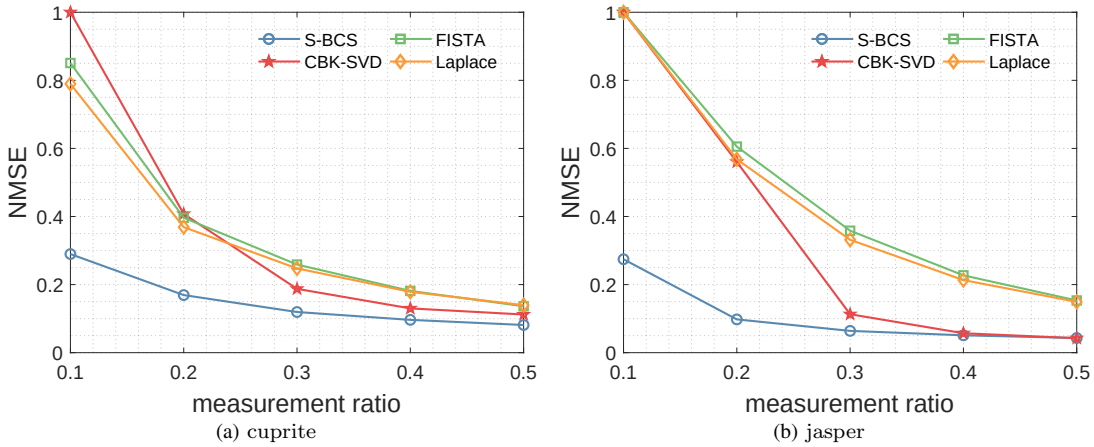
Datasets	# features	# samples
Cuprite	224	10000
Jasper	224	10000
SensIT	100	1500
USPS	256	3300
madelon	500	2600
isolet	617	1560

validation samples are randomly chosen for each dataset and the testing samples are accordingly chosen randomly from the rest of the dataset. To reduce the difference brought by different partitions of the validation and testing sets, we present the averaged results over five different runs for each dataset. For avoiding the exhaustive search for the optimal parameters, the additional regularization parameter λ in our method is empirically set as $\lambda = 1$, which implies a trade-off that the two terms, i.e., the error term of the compressive measurements $\sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{\Phi}_i \mathbf{x}_i\|_2^2$ and the penalty term of dictionary representation $\|\mathbf{X} - \mathbf{DS}\|_F^2$, are of the same importance for the recovery of $\mathbf{X}$. Besides, we set $\beta \geqslant 10^{-4}$ to prevent abrupt change of $\Delta\mathbf{D}$. Besides, for CS and S-BCS algorithms, we consider the commonly used DCT as the prior sparsity basis if not additionally mentioned. The measurement matrices involved are generated via the sparse Bernoulli distribution [10] for storage saving. Normalized mean squared error (NMSE) is used to evaluate the reconstruction performance,

$$\text{NMSE} = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}{\|\mathbf{x}_i\|_2^2} \tag{C1}$$

Appendix C.1 Hyperspectral Datasets

In this subsection, we compare the reconstruction performance of the algorithms on hyperspectral datasets. In the first experiment, we evaluate the performance of S-BCS under various measurement ratios. We randomly choose $n_{val} = 100$ and $n_{tst} = 2000$ samples from each hyperspectral dataset as the validation and testing sets respectively. We repeat such partition five times and Figure C1 shows the NMSEs averaged over five runs on these testing sets with measurement ratios varying from 0.1 to 0.5. We can see that S-BCS always outperforms other algorithms with varying measurement ratios. In particular, even with very small measurement ratio $p/d = 0.1$, S-BCS still obtains favorable performance while other algorithms totally fail to recover the data. In fact, using as much information as possible is very important at very small compression while our method can exactly borrow strength from both prior sparsity basis and data.



(a) cuprite (b) jasper

Figure C1 Comparisions on hyperspectral datasets with different measurement ratios.

From the view point of BCS, the above experiment shows that using prior knowledge about the sparsity basis indeed helps data recovery. However, whether the performance improvement of S-BCS relies heavily on the choice of prior sparsity basis has not been discussed. In the next experiment, we evaluate the prior sensitivity of S-BCS with respect to four different sparsity bases, including DCT, DWT using Haar basis and length-4 Daubechies basis, and random matrix generated from Gaussian distribution. The validation and testing sets are still chosen as the above experiment. Figure C2 shows the averaged NMSEs of S-BCS over five runs on different prior sparsity bases with varying measurement ratio. From Figure C2, we see that S-BCS relies more on the prior sparsity basis at smaller measurement ratio while gains comparable performance on all

four sparsity bases at larger measurement ratio. Such behavior can be explained by the fact that when the measurement ratio is sufficiently large, the available information from the measurements is enough to learn an appropriate sparsity basis whereas for smaller measurement ratio, the prior information plays a bigger role to compensate the lack of sufficient measurements.

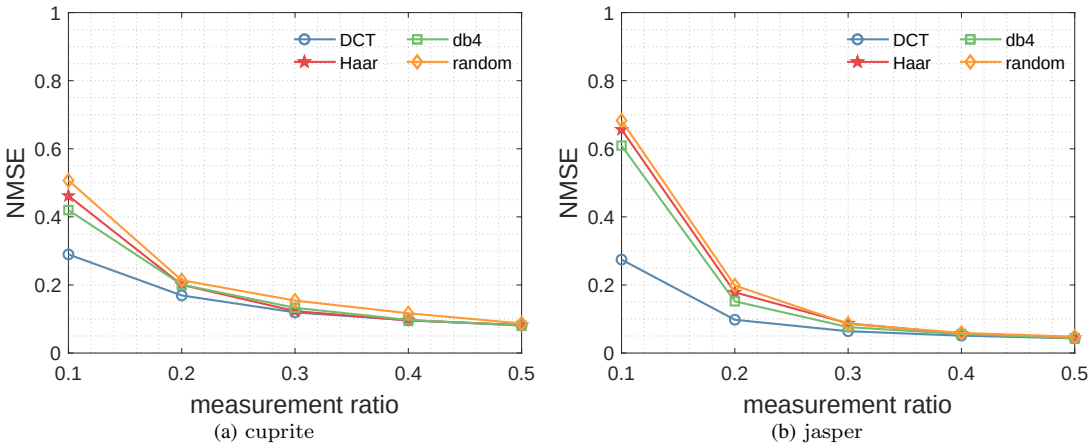


Figure C2 Results on different prior sparsity bases with different measurement ratios.

In [11], it has been demonstrated that the success of compressive dictionary learning depends on sufficient samples whereas the bound on n is less practical. Thus, we empirically examine the performance of S-BCS on different sample sizes. Specifically, we randomly choose validation set with fixed size $n_{val} = 100$ and testing set with varied size $n_{tst} = \{100, 200, 500, 1000, 2000, 5000\}$. The measurement ratio p/d is fixed as 0.3. Figure C3(a) and C3(b) show the averaged NMSEs of the algorithms over five runs on each dataset respectively. Meanwhile, in order to intuitively compare the computational complexity of these methods, we present the CPU time of these methods against the corresponding sample sizes in Figure C3(c) and C3(d). It is obvious that the increasing number of data samples brings performance gain for the BCS while no effect on CS. Besides, S-BCS can achieve smaller recovery error than CS by using much smaller number of samples (e.g. $n = 100$) than CBK-SVD required (e.g. $n > 1000$). Such superiority makes it possible to adapt S-BCS to online tasks with mini-batch. Moreover, our method achieves higher computational efficiency than BCS especially for relatively larger sample sizes.

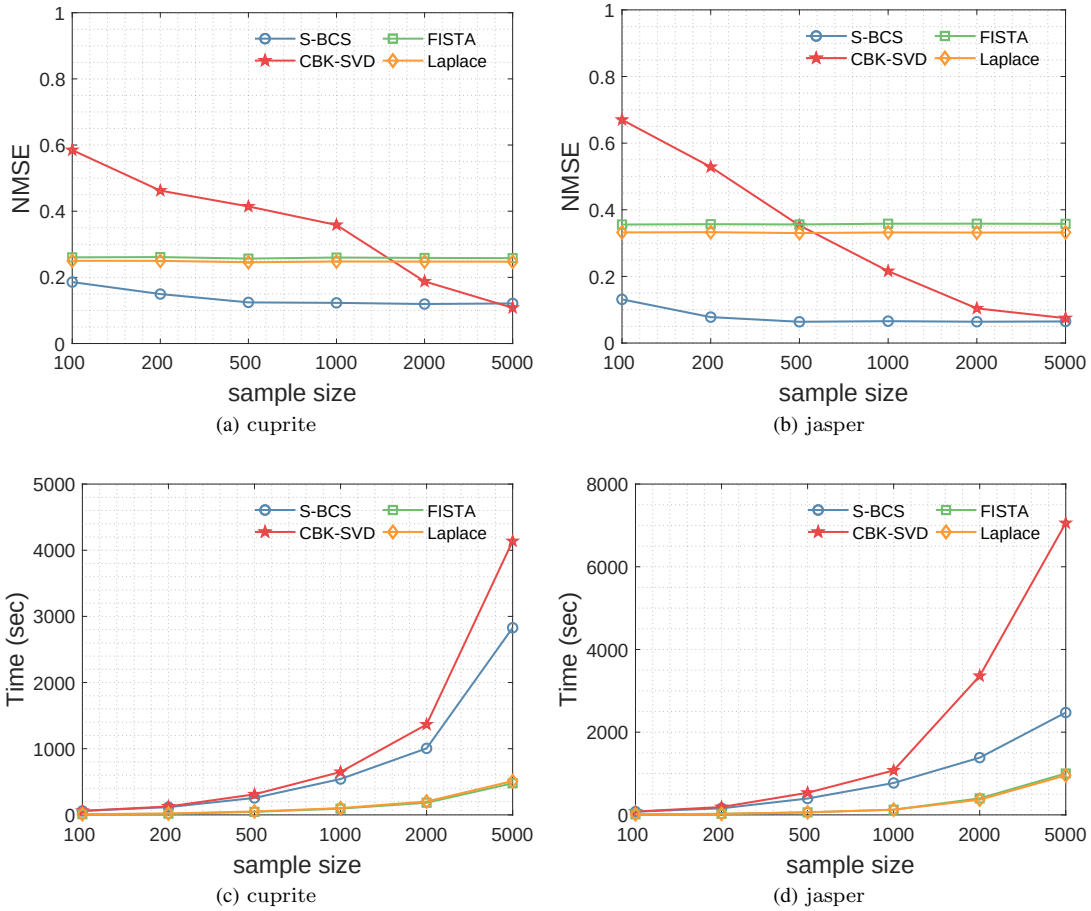
To evaluate the ability of the one-pass online version, OS-BCS, we compare its performance with S-BCS on different batch sizes. The sample sizes of validation and testing are fixed as $n_{val} = 100$ and $n_{tst} = 2000$ respectively, the measurements ratio p/d is fixed as 0.3. Figure C4 shows the averaged NMSEs of OS-BCS over five runs with the batch size varying from $\{1, 5, 10, 50, 100\}$. We see from Figure C4 that with increasing number of visited samples, the NMSE of S-BCS smoothly decreases on all batch sizes. This experimentally reflects the convergence of OS-BCS. Besides, with increasing batch size, the NMSE of OS-BCS is comparable to that of S-BCS.

The above experiments verify the superiority of the semi-blind learning strategy combined with FISTA. To ensure a convincing experimental verification on such superiority, we conduct the following experiments to show the recovery performance of another five commonly-used CS algorithms, OMP [12], CoSaMP [13], L1LS [14], SpaRSA [15] and Homotopy [16], with or without semi-blind learning. According to Figure C3(a) and Figure C3(b), sample size $n_{tst} = 1000$ is enough to ensure good performance of S-BCS while continuing to increase will not bring obvious improvement. Hence, in this experiment, we fix $n_{val} = 100$ and $n_{tst} = 1000$ as the sample sizes of the validation and testing sets respectively. The measurement ratio is fixed as $p/d = 0.3$. Figure C5 shows the averaged NMSEs of the algorithms over five runs on each dataset respectively. From the results, we find that the semi-blind learning strategy enjoys the same superiority when combined with other CS algorithms.

Appendix C.2 Other Public Datasets

In this subsection, we conduct experiments on the other four public datasets to examine the reconstruction performance of S-BCS on different types of data. For each dataset, the number of dictionary atoms is chosen as around one fifth of the data dimensionality. On one hand, such choice can reduce the computational complexity. On the other hand, it has been demonstrated that the more atoms we choose, the more data samples required by BCS for successful recovery [11]. The sample size of the validation set is fixed as $n_{val} = 100$ and the testing set consists of the rest data samples. Figure C6 shows the NMSEs of the algorithms averaged over five runs with various measurement ratios. Clearly, S-BCS still outperforms the other algorithms on different data types used here.

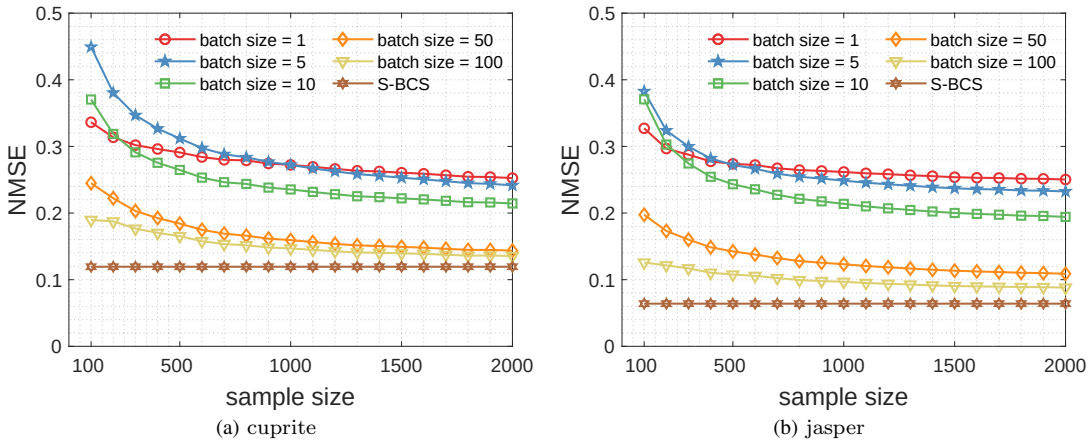
To further evaluate our method, we visualize some of the recovery results of USPS dataset and the corresponding dictionaries learned/used for the compared algorithms in figure C7 and figure C8 respectively. The measurement ratio is fixed as $p/d = 0.5$. As can be seen from figure C7, both BCS algorithms yield better visual effect than CS algorithms while



(a) cuprite (b) jasper

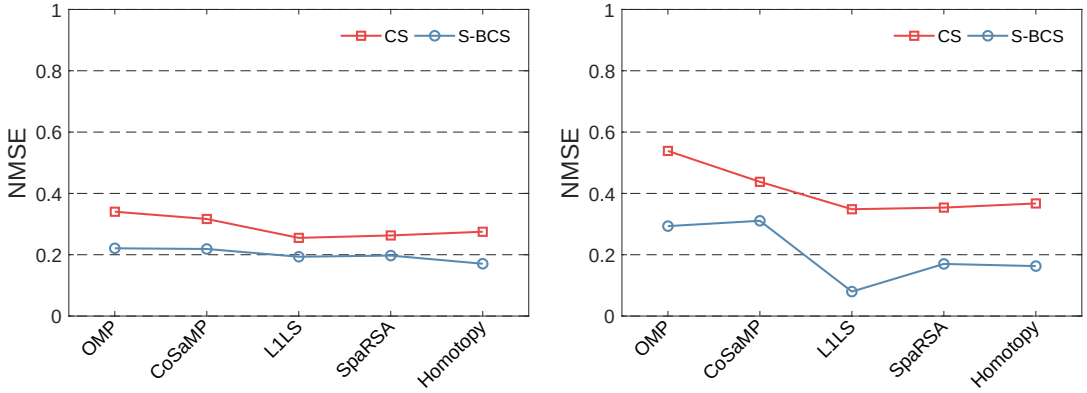
(c) cuprite (d) jasper

Figure C3 Comparisons on hyperspectral datasets with different sample sizes.



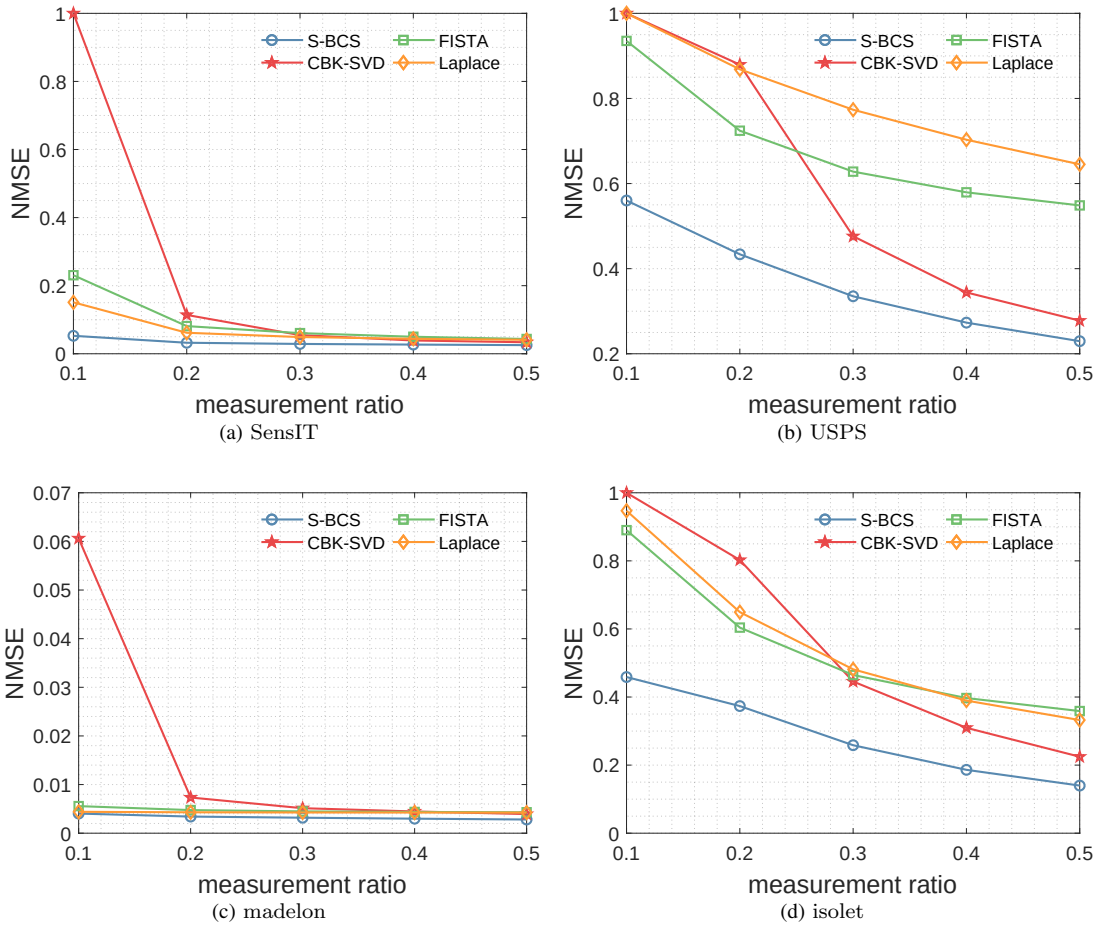
(a) cuprite (b) jasper

Figure C4 Results of OS-BCS with different batch sizes.



(a) cuprite (b) jasper

Figure C5 Comparisons on hyperspectral datasets with different sample sizes.



(a) SensIT (b) USPS

(c) madelon (d) isolet

Figure C6 Comparisions on four public datasets with different measurement ratios.

our method produces visually clearer recovered images than CBK-SVD. By comparing subfigure C8(a) with subfigures C8(b) and C8(c), we see that S-BCS has adaptively transferred some prior dictionary atoms to specific digit recovery task (as indicated by the red box in subfigure C8(a)) while preserving the rest almost unchanged (as indicated by the green box in C8(a)). In fact, such results also verify the sparsity of $\mathbf{S}$ since the zero/negligible-valued rows in $\mathbf{S}$ tend to make the corresponding columns of $\Delta\mathbf{D}$ zero, implying that these columns in $\mathbf{D}$ remain almost unchanged. Moreover, we gain an insight that the dictionary transfer occurs much more on low frequency bases (as shown by the red box in subfigure C8(c)) than high frequency bases (as shown by the green box in subfigure C8(c)), which is basically consistent with a fact that the high frequency components of most images under our consideration are indeed fewer than their low frequency counterparts. It is such a mechanism that our algorithm can naturally adapt the low frequency bases to specific tasks, where such adaptively transferred bases still comprise most information of the images.

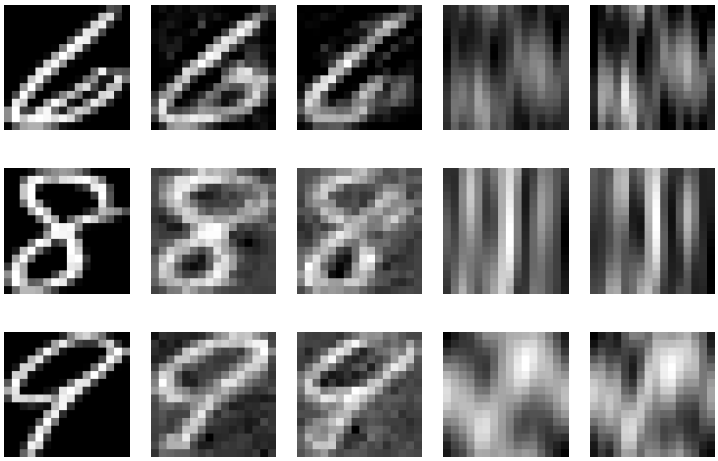


Figure C7 Visual comparisons on USPS datasets. The first column is the ground-truth images of digits '6', '8' and '9', the other columns are the recovered images corresponding to S-BCS, CBK-SVD, FISTA and Laplace, respectively.

For further supporting such an insight, we conduct an experiment on another visual dataset, Caltech-101, from which we choose two classes of vehicles, i.e., motorbikes and airplanes, that totally contains 1598 images. Following, we resize these images into 16×32 pixels and refer to this subset as Caltech-2. Figure C9 shows the dictionary learned by S-BCS on Caltech-2 and its prior dictionary. From figure C9, we can gain a similar insight to the previous experiment except that the number of the transferred low frequency bases is also task-dependent. Specifically, since the individual differences between images from Caltech-2 are smaller than those from USPS (e.g., the airplanes from Caltech-2 look more like each other than the digits of '6' from USPS), thus the images from Caltech-2 are expected to be represented via fewer dictionarie atoms than those from USPS.

The above experiments verify our insight from the viewpoint of data itself, i.e., adopting the same prior sparsity basis for different datasets. Next, we investigate from another viewpoint that adopts different prior sparsity bases for the same dataset. Concretely, we perform experiments on another two prior sparsity bases, DWT using Haar and db4 bases respectively. Figure C10 shows the dictionaries learned by S-BCS on USPS dataset. This result further supports our insight that the low-frequency bases have been transferred to specific task while most high-frequency bases still remain nearly unchanged.

The insights from the above three experiments show that S-BCS can adapt the prior sparsity basis to a task-dependent one that comprises both the frequency characteristic of images and the task-dependent information for specific tasks. In contrast, CS and BCS can not enjoy such a mechanism since the dictionary used/learned by them can only reflect the frequency characteristic and task-dependent information respectively. In this sense, our model achieves greater interpretability. Consequently, the dictionary learned by S-BCS indeed builds a bridge at middle of the 'spectrum', where the two ends of the 'spectrum' are the dictionary used/learned by CS and BCS respectively.

References

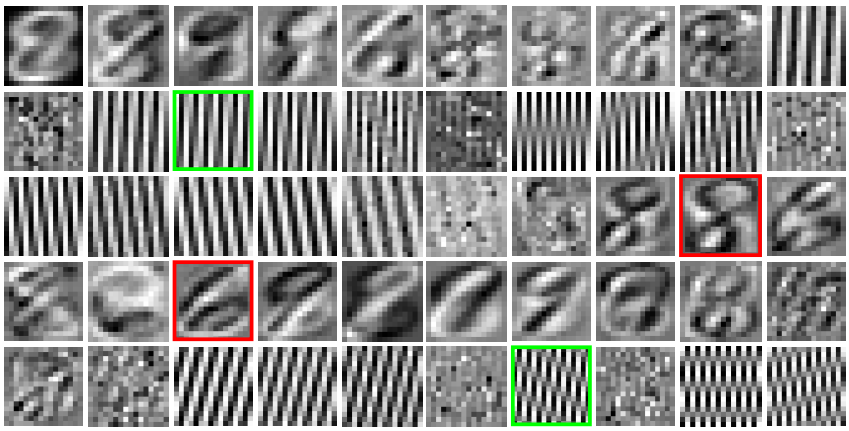
1 Feng J S, Xu H, and Yan S C. Online robust PCA via stochastic optimization. In: Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, 2013. 404–412

2 Fowler J E. Compressive-projection principal component analysis. IEEE Trans. Image Processing, 2009, 18:2230–2242

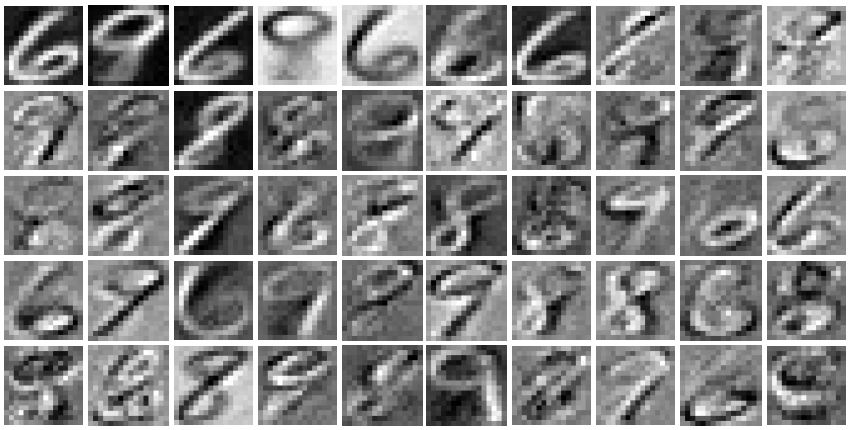
3 Duarte M F and Hu Y H. Vehicle classification in distributed sensor networks. Journal of Parallel and Distributed Computing, 2004, 64:826–838

4 Hull J J. A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Machine Intell, 1994, 16:550–554

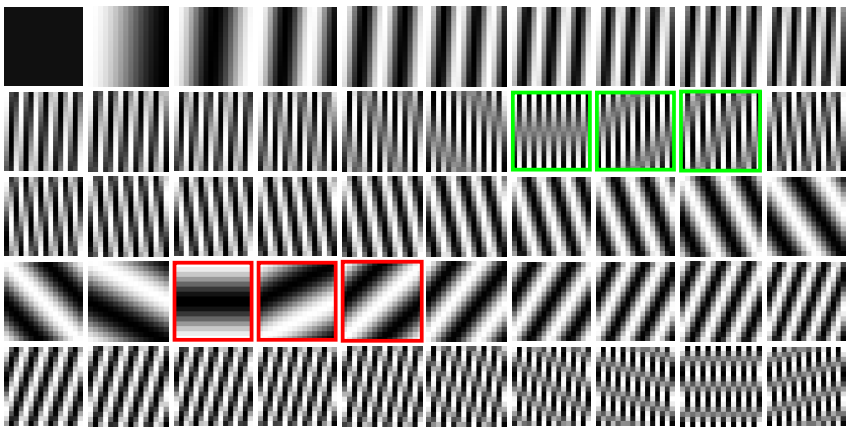
5 Guyon I, Gunn S, Ben-Hur A, et al. Result analysis of the NIPS 2003 feature selection challenge. In: Advances in Neural Information Processing Systems, Vancouver, Canada, 2005. 545–552



(a) Dictionary learned by S-BCS

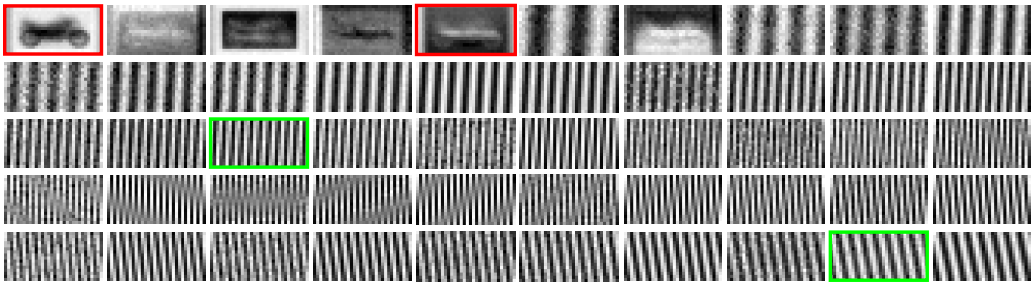


(b) Dictionary learned by CBK-SVD

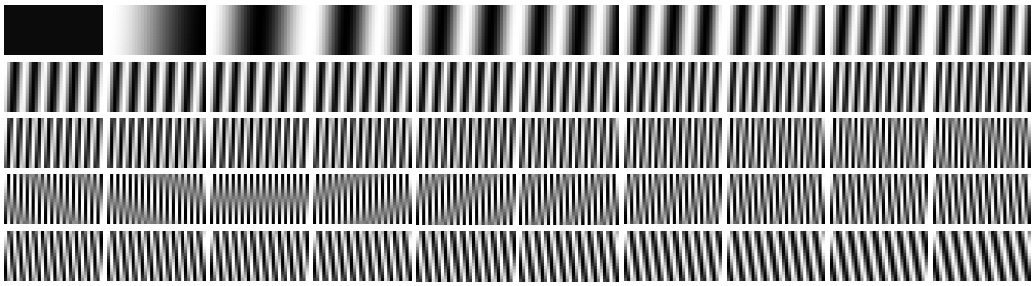


(c) DCT

Figure C8 Comparisons on the dictionaries learned/used for USPS dataset.

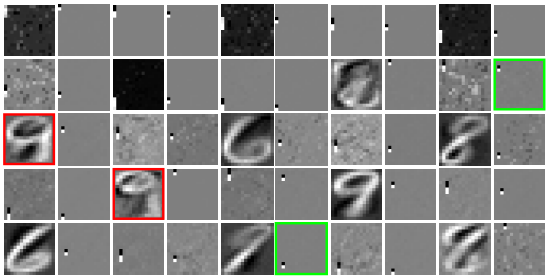


(a) Dictionary learned by S-BCS

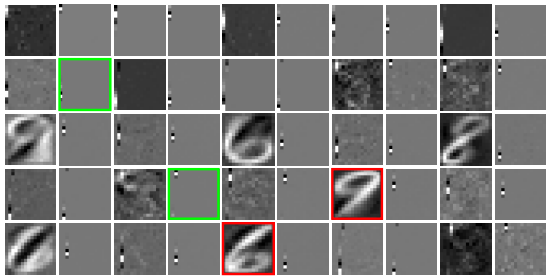


(b) DCT

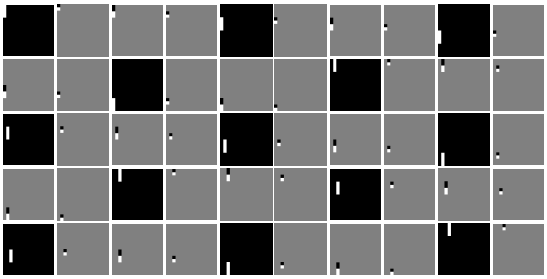
Figure C9 Comparisons on the dictionaries learned/used for Caltech-2 dataset.



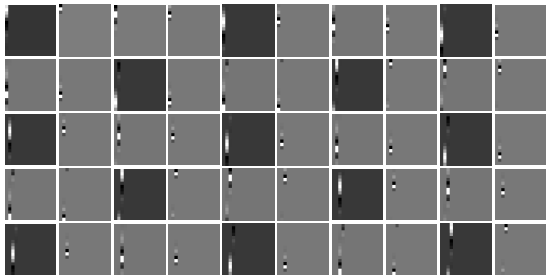
(a) Dictionary learned by S-BCS using Haar



(b) Dictionary learned by S-BCS using db4



(c) DWT using Haar basis



(d) DWT using db4 basis

Figure C10 Comparisons on the dictionaries learned/used for USPS dataset.

6 Dietterich T G and Bakiri G. Error-correcting output codes: A general method for improving multiclass inductive learning programs. AAAI, 1991. 572–577

7 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2009, 2:183–202

8 Babacan S D, Molina R, and Katsaggelos A K. Bayesian compressive sensing using Laplace priors. IEEE Trans. Image Processing, 2010, 19:53–63

9 Testa M, Magli E. Compressive Bayesian K-SVD. Signal Processing: Image Communication, 2018, 60:1–5

10 Pourkamali A F, Becker S, and Hughes S M. Efficient dictionary learning via very sparse random projections. In: Proceedings of International Conference on SampTA, Washington, DC, USA, 2015. 478–482

11 Aghagolzadeh M, Radha H. New guarantees for blind compressed sensing. In: Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2015. 1227–1234

12 Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inform. Theory, 2007, 53:4655–4666

13 Deanna Needell and Joel A Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Communications of the ACM, 2010, 53:93–100

14 Kim S, Koh K, Lustig M, et al. An interior-point method for large-scale ℓ_1-regularized least squares. IEEE J. Sel. Top. Sign. Proces, 2007, 1:606–617

15 Wright S J, Nowak R D, and Figueiredo M A T. Sparse reconstruction by separable approximation. IEEE Trans. Signal Processing, 2009, 57:2479–2493

16 Malioutov D M , Cetin M, Willsky A S. Homotopy continuation for sparse signal representation. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, 2005. 733–736