IK-SVD:
Dictionary Learning for Spatial Big Data via Incremental Atom Update

Lizhe Wang | Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China
Ke Lu | University of the Chinese Academy of Sciences, Beijing, China
Peng Liu | Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China
Rajiv Ranjan | CSIRO Computational Informatics, Canberra, Australia
Lajiao Chen | Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China

Big Data, a large and complex collection of datasets characterized by four V's (volume, variety, veracity, and velocity), is difficult to deal with using traditional data processing algorithms and models. A proposed dictionary learning algorithm, which extends the classical method that uses the K-means and Singular Value Decomposition (K-SVD) algorithm by incrementally updating atoms, will ably represent the spatiotemporal remote sensing of Big Data and do so both efficiently and sparsely.

Big Data is a collection of datasets so large and complex that it's difficult to work with using traditional data processing algorithms and models. The challenges include data acquisition, storage, search, sharing, transfer, analysis, and visualization. Scientists regularly encounter limitations due to large datasets in many areas, such as geosciences and remote sensing, complex physics simulations, and environmental research. In remote sensing applications, the size of the dataset grows in part because data are increasingly being gathered by many different satellite sensors with different resolutions and different spectral characters; more importantly, the data are with high spatial and temporal resolution.

Big Data is difficult to deal with using traditional methods. How to represent a big dataset is a fundamental problem in Big Data research, as most data processing tasks rely on an appropriate data representation. For many tasks, such as sampling, reconstruction, compression, retrieval, communication, classification, and so on, a sparse data representation is preferable. And for remote-sensing Big Data, sparseness is increasingly important for many algorithms such as image segmentation, image fusion, change detection, feature extraction, and image interpretation. We can sparsely represent the data by a basis set, that is, a dictionary. Most commonly, we'd use either an analytic dictionary or an unanalytic dictionary (see the "Related Work in Sparse Representation" sidebar for more details.)

Many recent algorithms for unanalytic dictionary learning are iterative batch procedures. They access the whole training set at each iteration and minimize the cost function under some constraints. However, these algorithms can't deal efficiently with very large datasets, or dynamic data changing over time. To address these issues, researchers proposed the popular Online Dictionary Learning (ODL) method.[1] Another, competitive method is the Recursive Least Squares Dictionary Learning (RLSDL) algorithm.[2] Both ODL and RLSDL have the ability to train large data sets. ODL and RLSDL, in theory, are not dictionary methods specialized to a particular (small) training set, however, they may encounter some problems while dealing with real remote-sensing Big Data. First, ODL and RLSDL update all atoms for every new sample, which may be unrealistic given the many atoms in a truly big dataset; second, the fixed number of atoms in the dictionary learning process isn't very adaptable. For the dictionary learning of a big dataset, the process should be dynamic given how many atoms there are and which atoms need to be updated.

 Copublished by the IEEE CS and the AIP 1521-9615/14/$31.00 © 2014 IEEE

Related Work in Sparse Representation

Earlier research studies on sparse representation focused on *analytic* dictionaries. For example, the Fourier dictionary was for smooth functions, whereas the wavelet dictionary was for piecewise-smooth functions with point singularities. Recently, Emmanuel Candès and his colleagues proposed the idea of curvelet transform,[1] whereby each curvelet atom associates with a specific location, orientation, and scale, which together make the atom efficiently represent the smooth curves. The bandelet transform[2] represents one of the most recent contributions in the area of signal-adaptive transforms. Some other adaptive analytic dictionaries, such as directionlet transform[3] and grouplet transform,[4] which are also popular in sparse representation research areas.

Another large branch is the *un-analytic* dictionary. Unlike decompositions based on a predefined analytic base (such as wavelet) and its variants, we can also learn an overcomplete dictionary without analytic form, which neither have fixed forms of atoms nor require orthogonal basis vectors. The basic assumption of the learning approach is that we can extract the structure of complex incoherent characters directly from the data rather than by using a mathematical description. The *Method of Optimal Directions* (MOD)[5] is one of the earliest unanalytical methods. Another un-analytic method is *Generalized Principal Component Analysis* (GPCA),[6] in which represent each sample by only one of the subspaces. K-SVD[7] focuses on the same sparsification problem as the MOD and employs a similar block-relaxation approach. K-SVD's main contribution is its method of updating atoms. Rather than using a matrix inversion, K-SVD updates atoms one at a time in a simple and efficient process. Nonparametric Bayesian dictionary learning,[8] another un-analytic method, employs a truncated beta-Bernoulli process to infer an appropriate dictionary, and it obtains significant improvements in image recovery.[8]

References

1. E.J. Candès et al., "Fast Discrete Curvelet Transforms," *Multiscale Modeling and Simulation*, vol. 5, no. 3, 2006, pp. 861–899; doi: 10.1137/05064182X.
2. E. LePennec and S. Mallat, "Sparse Geometric Image Representations with Bandelets," *IEEE Trans. Image Processing*, vol. 14, no. 4, 2005, pp. 423–438.
3. V. Velisavljevic et al., "Directionlets: Anisotropic Multidirectional Representation with Separable Filtering," *IEEE Trans. Image Processing*, vol. 15, no. 7, 2008, pp. 1916–1933.
4. S. Mallat, "Geometrical Grouplets," *Applied and Computational Harmonics Analysis*, vol. 26, no. 2, 2009, pp. 161–180.
5. K. Engan, S.O. Aase, and J. Hakon Husoy, "Method of Optimal Directions for Frame Design," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 5, 1999, pp. 2443–2446.
6. Y. Ma, R. Vidal, and S. Sastry, "Generalized Principal Component Analysis (GPCA)," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, 2005, pp. 1945–1959.
7. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, 2006, pp. 4311–4322.
8. M. Zhou et al., "Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images," *IEEE Trans. Image Processing*, vol. 21, no. 1, 2012, pp. 130–144.

Here, we propose the Incremental K-SVD (IK-SVD) algorithm, which yields dynamic dictionaries by sequentially updating dictionary atoms one at a time. Furthermore, when the number of atoms changes with the training process, the dictionary is able to represent spatiotemporal remote-sensing Big Data efficiently and sparsely.

Dictionary Learning for Finite Training Datasets

Classical dictionary learning techniques for sparse representation that consider a data sample $y \in R^h$ can be described as $y = D\alpha$, where $D \in R^{h\times n}$ is a dictionary with n atoms, and $\alpha \in Rn$ is the coefficients for the sparse representation. We typically consider the case $n > h$, suggesting that the dictionary is redundant. The number of nonzero coefficients in the representation is denoted as $k = \|\alpha\|_0$, where k is expected to be very small (see Table 1). $Y = D\alpha$ implies that the sample y can be characterized as a linear combination of a few columns from the dictionary $D \in R^{h\times n}$, which is also referred as to the set of atoms. Then, the problem is

$$\min_{D,\alpha} = \|y - D\alpha\|_2^2 \quad \text{subject to} \quad \|\alpha\|_0 \le k. \tag{1}$$

In another expression of the object function and constraints, we can also control the error of the reconstruction as $\|y - D\alpha\|_2^2 \le \sigma$. Usually, there's a group of samples that need to be represented, so it's denoted as $Y = \{y_1, \ldots, y_r\}$, and the coefficients set is denoted as $X = \{\alpha_1, \ldots, \alpha_r\}$, where $X \in R^{n\times r}$. Now, the sparse representation and dictionary learning problem is defined as

$$\min_{D,\alpha_1,\cdots,\alpha_r} \sum_{i=1}^{r} \|y - D\alpha\|_2^2 + \lambda \sum_{i=1}^{r} \|\alpha\|_0. \tag{2}$$

It's equal to

$$\min_{D,X} \|Y - DX\|_2^2 + \lambda \|X\|_0 , \quad (3)$$

where $\|\cdot\|_2$ is L_2 norm, $\|\cdot\|_0$ is L_0 norm, and λ is a regularization parameter. It's well-known that L_0 regularization yields a sparse solution for X, which scientists have proved earlier.[3] This problem to search X is also known as the Lasso or basis pursuit. To prevent the dictionary D from having arbitrarily large values (which would lead to arbitrarily small values of α), it's common to constrain its atoms $D = \{d_1, \ldots, d_n\}$ to have an $L_2 - norm$ less than or equal to one.[4] The problem of minimizing the object function in Equation 3 is a joint optimization problem with respect to the dictionary D and the coefficients X in the sparse decompositions. The object function isn't jointly convex, but it's convex with respect to each of the two variables D and X when the other one is fixed.

There's a large body of research[1,2,4,5] that focuses on how to find a good dictionary by training samples and how to represent a small dataset $Y = \{y_1, \ldots, y_r\}$ sparsely. Under most circumstances, there are two stages in the dictionary leaning algorithms: one is the sparse coding stage, which searches for the optimal solution of Equation 4,

$$\min_{\alpha} \|Y - D\alpha\|_2^2 + \lambda \|\alpha\|_0 . \quad (4)$$

With a fixed dictionary D, the sparse coding problem of Equation 4 is an L_0 regularized problem. It can be solved using many methods.[6,7] The other problem is the atoms-updating stage, which means finding the solution of Equation 5:

$$\min_{D} \|Y - D\alpha\|_2^2. \quad (5)$$

The atoms-updating stage in Equation 5 is the main characteristic that distinguishes many different popular dictionary learning methods. The ODL and RLSDL methods employ dynamic schemes in the atom-updating stage, so that they can, theoretically, train an infinite dataset. The classical K-SVD method has very good performance on small data, but it's impossible to perform K-SVD dictionary learning for Big Data because we could neither read all data samples into the computer memory nor perform Singular Value Decomposition (SVD) decomposition of a very large matrix.

However, K-SVD has its own advantages. Its dictionary update scheme is a good model with clear mathematics and physics meanings.

Table 1. Notation.

Symbol	Meaning
Y	Sample data
$Y = \{y_1, \ldots, y_r\}$	Small dataset
$\{Y_1, \ldots, Y_s\}$,	Large dataset
α	Coefficient vector
$\alpha_i = \{\alpha_i(1), \ldots, \alpha_i(n)\}$	α_i with n components
$X = \{\alpha_1, \ldots, \alpha_r\}$	Coefficients set
$\{X_1, \ldots, X_s\}$,	Large coefficients set
α_T^j	The jth row in X
d	Atom
$D = \{d_1, \ldots, d_r\}$	Atoms set
λ	Regularization parameter
$\omega_k = \left\{i \middle\| 1 \le i \le r, \alpha_T^k(i) \ne 0\right\}$	Nonzero support of α
E	Error matrix

To sparsely represent the big dataset from remote sensing, we'll extend the K-SVD algorithm and explore the redundant features of a spatiotemporal remote-sensing image set.

In this article, we propose the incremental K-SVD algorithm, which can introduce new atoms into the dictionary and update the small part of the dictionary by using only the current sample set, step by step in each iteration. We'll discuss in detail how to train a dictionary from a large spatiotemporal dataset by our algorithm.

Dictionary Learning by IK-SVD

We've already analyzed the classical K-SVD method, which is applicable to small datasets. Now assume that there's a big dataset $\{Y_1, \ldots, Y_s\}$, where s means a different time or a different location. These multi-spatiotemporal data share similar features and differences. Since redundancy information always exists in a large spatiotemporal dataset $\{Y_1, \ldots, Y_s\}$, it's possible to represent it sparsely by dictionary learning.

Most traditional methods aren't applicable to every large dataset. This means that we can't train all the samples in a big dataset $\{Y_1, \ldots, Y_s\}$ at one time to get the final dictionary D. In a learning algorithm for a big dataset, both the number of atoms and the samples to be used should change dynamically. Therefore, the problem becomes this:

If for the dataset $\{Y_1, \ldots, Y_s\}$ we've already obtained its dictionary $Ds = \{d_1, \ldots, d_n\}$, then for the next scene of image Y_{s+1} with the index $s + 1$, we need to find a new dictionary $D_{s+1} = \{d_1, \ldots, d_{n+1}, \ldots, d_{n+m}\}$, which has m more atoms as $d_{n+1} \ldots, d_{n+m}$ added, and is able to sparsely represent, the dataset $\{Y_1, \ldots, Y_{s+1}\}$.

Obviously, we hope that the latest atoms $d_1, \ldots, d_n$ are still reserved in the new dictionary $\{d_1, \ldots, d_n, d_{n+1}, \ldots, d_{n+m}\}$. We also hope that, when every new data subset Y_{s+1} is trained, only a few new atoms are added to D_{s+1}, so that we can efficiently and sparsely represent both $\{Y_1, \ldots, Y_s\}$ and Y_{s+1}. As a result, in the training process, we could obtain $\{X_1, \ldots, X_s, X_{s+1}\}$, which is the sparse coefficients matrix sequence for the dataset $\{Y_1, \ldots, Y_s, Y_{s+1}\}$. Since we already defined that D_s is a part of D_{s+1}, and D_s can already sparsely represent $\{Y_1, \ldots, Y_s\}$, we need only to update coefficients X_{s+1}, which relates to the sparse representation for Y_{s+1} based on dictionary D_{s+1}. Now we define the new object function for the incremental learning model as

$$\min_{D_{s+1}, X_{s+1}} \left\| Y_{s+1} - D_{s+1} X_{s+1} \right\|_2^2 + \lambda \left\| X_{s+1} \right\|_0. \quad (6)$$

When training data $Y_1, \ldots, Y_s, Y_{s+1}$, we assume that they have the same number of r samples, and then we have

$$Y_{s+1} = \{y_1, \ldots, y_r\}. \quad (7)$$

Since there are r samples in each Y_{s+1}, the corresponding coefficient X_{s+1} is

$$X_{s+1} = \{\alpha_1, \ldots, \alpha_r\}. \quad (8)$$

In Equation 8, the coefficient vector α_i with more components, where $1 \leq i \leq r$, becomes

$$\alpha_i = \{\alpha_i(1), \ldots, \alpha_i(n), \alpha_i(n+1), \ldots, \alpha_i(n+m)\} \quad (9)$$

We see that there are more components, such as $\alpha i (n + 1), \ldots, \alpha_i (n + m)$, in the coefficient vector αi in Equation 9, because there are more atoms in the current dictionary D_{s+1}. Since we can't train all the samples at one time, we construct and update every small group of atoms for every new training set Y_{s+1}.

Actually, we care more about the current atoms $d_{n+1}, \ldots, d_{n+m}$ and their coefficients. In an extreme case, for the current Y_{s+1}, if $\alpha_i(1), \ldots, \alpha_i(n)$ in every coefficient vector α_i are efficient and sparse enough, even $d_{n+1}, \ldots, d_{n+m}$ aren't necessary, and $\alpha_i (n+1), \ldots, \alpha_i (n+m)$ can all be zero. However, usually there are new atoms, such as $d_{n+1}, \ldots, d_{n+m}$, that need to be added and updated. This is because, for a large spatiotemporal dataset, there are always some image features in Y_{s+1} that can't be efficiently represented by atoms trained from $\{Y_1, \ldots, Y_s\}$.

When we solve Equation 6, following the idea of classical K-SVD, the jth row in X_{s+1} is denoted as α_T^j (this isn't the vector α^j, which is the jth column in X). For an arbitrary new kth atom, the first term of the object function in Equation 6 can be denoted as

$$\left\| Y_{s+1} - D_{s+1} X_{s+1} \right\|_2^2 = \left\| Y_{s+1} - \sum_{j=1}^{n} d_j \alpha_T^j - \sum_{j=n+1}^{k-1} d_j \alpha_T^j - \sum_{j=k+1}^{n+m} d_j \alpha_T^j - d_k \alpha_T^k \right\|_2^2. \quad (10)$$

It is the changing form of the object function of the proposed incremental dictionary learning. In Equation 10, there are two obvious differences from the classical K-SVD model: one is that the current sample Y_{s+1} and the old atoms $d_1, \ldots, d_n$ trained by old samples are linked and combined into one object function; the other is $n + 1 \leq k \leq n + m$, which means that for the new training samples Y_{s+1}, we'll update only the new atoms within $d_{n+1}, \ldots, d_{n+m}$.

Therefore, the equation changes to

$$\left\| Y_{s+1} - D_{s+1} X_{s+1} \right\|_2^2 = \left\| E_{s+1}^k - d_k \alpha_T^k \right\|_2^2, \quad (11)$$

where

$$E_{s+1}^k = Y_{s+1} - \sum_{j=1}^{n} d_j \alpha_T^j - \sum_{j=n+1}^{k-1} d_j \alpha_T^j - \sum_{j=k+1}^{n+m} d_j \alpha_T^j. \quad (12)$$

We've decomposed the multiplication $D_{s+1} X_{s+1}$ into the sum of $n + m$ matrices. Among those $n + m$, $n + m - 1$, terms are assumed fixed, and one (the kth) remains in question. However, it's different from the traditional K-SVD method: for the new training sample data Y_{s+1}, we'll never update atoms of $d_1, \ldots, d_n$ that are already trained by $\{Y_1, \ldots, Y_s\}$. Every time, what we'll update are only atoms of $d_{n+1}, \ldots, d_{n+m}$.

Therefore, for Y_{s+1}, we calculate only the current error matrix E_{s+1}^k, where $n + 1 \leq k \leq n + m$. The meaning of E_{s+1}^k differs from that in the work of Michael Aharon and his colleagues.[4] The proposed matrix E_{s+1}^k stands for the error for the current samples Y_{s+1} but not all history samples,

when atoms $d_1, \ldots, d_n$ are fixed and when the atom d_k is removed, where $n + 1 \leq k \leq n + m$. Respectively, we also need to define a new ω_k for new atoms as

$$\omega_k = \left\{ i \middle| 1 \leq i \leq r, \alpha_T^k(i) \neq 0 \right\}, \tag{13}$$

where $n + 1 \leq k \leq n + m$.

Note that the error matrix E_{s+1}^k stands for how well the dictionary D_{s+1} without dk can represent the current training data Y_{s+1}, and the information from known atoms $d_1, \ldots, d_n$ is still associated with E_{s+1}^k. On the other hand, the initial value of the D_{s+1} for the proposed method also differs from that of the traditional K-SVD, which we'll discuss next.

Estimate the New Atoms' Initial Value

Although we can add new atoms to the current dictionary, it's still very difficult to set the initial value of the new atoms when a batch of new samples is introduced into the training process. If the old dictionary D_s can efficiently and sparsely represent the new samples Y_{s+1}, we don't need to create new atoms and put them into D_{s+1}. However, there are often new image features from the new samples, which can't be efficiently represented by old atoms, so we often need to add new atoms. We'll select special samples as the initial value of the new atoms. If we set improper initial values for the new atoms, the training process will be slow and inefficient. Therefore, it's very important to make a good choice of new atoms for incremental dictionary learning.

When considering each new Y_{s+1}, we first perform a sparse coding for Y_{s+1} using dictionary D_s to evaluate how well the old dictionary D_s could represent the current samples Y_{s+1}. Then, there's

$$\min_{X_s} \left\| Y_{s+1} - D_s \mathbf{X}_s \right\|_2^2 + \lambda \left\| \mathbf{X}_s \right\|_0. \tag{14}$$

We call Equation 14 the initial representation. In this initial representation, for an arbitrary coefficient α_i vector within $\mathbf{X}_s$, it has n components as Equation 15 but not $n + m$ components:

$$\alpha_i = \{\alpha_i(1), \ldots, \alpha_i(n)\}. \tag{15}$$

The coefficient α_i characterizes the relationship between new samples Y_{s+1} and the old atoms D_s. To utilize the sparse coefficients to assist in introducing new atoms, we use the idea of active learning to set the initial value for new atoms.[8] The basic idea of active learning is to iteratively enlarge the training set by requesting an expert to label new samples from the unlabeled set in each iteration.[9] Here, we propose to use the entropy of information theory to decide which new samples will be the initial value of new atoms.

First, we select the samples from Y_{s+1} whose coefficients aren't sparse enough when we solve Equation 14. Then, among all the samples that can't be sparsely represented by old atoms, we need to select the samples showing maximal disagreement between the different atoms, which will be the initial value of the new atom. It's a little similar to the active learning scheme, which employs a Mutual Information (MI)-based criterion.[8] The difference is that we don't label the new sample but treat it as a new atom. Now, we define the new atom d_{new} as

$$d_{\text{new}} = \max_{\alpha_i \in X_{s+1}} H(\alpha_i), \tag{16}$$

where

$$H(\alpha_i) = \sum_{j=1}^{n} p(l = d_j | \alpha_i) \log(p(l = d_j | \alpha_i)), \tag{17}$$

and where

$$p(l = d_j | \alpha_i) = \frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)}. \tag{18}$$

Actually, $H(\alpha_i)$ also is

$$H(\alpha_i) = \sum_{j=1}^{n} \frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)} \log \left(\frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)} \right). \tag{19}$$

In iterating our proposed method, we first select a group of samples that can't be sparsely represented by the old dictionary. Then we calculate their entropy by Equation 17 and select m samples with the largest entropy as the initial value of the new atoms.

Now, we summarize the proposed dictionary learning algorithm as follows:

1. The big dataset is $\{Y_1, \ldots, Y_s, \ldots, Y_S\}$, where $1 \leq s \leq S$. Train the sample subset Y_1 by classical K-SVD, and get the initial dictionary $D_1 = \{d_1, \ldots, d_n\}$. Set $s = 2$, and $J = 1$.
2. Solve object function (14), select m samples based on Equation 16, and $D_s^{(J)} = D_{s-1} \bigcup \left\{ d_{n+1}, \ldots, d_{n+m} \right\}$.
3. For the sparse coding stage, use the Orthogonal Matching Pursuit (OMP) algorithm to compute the representation Y_s by the solution of

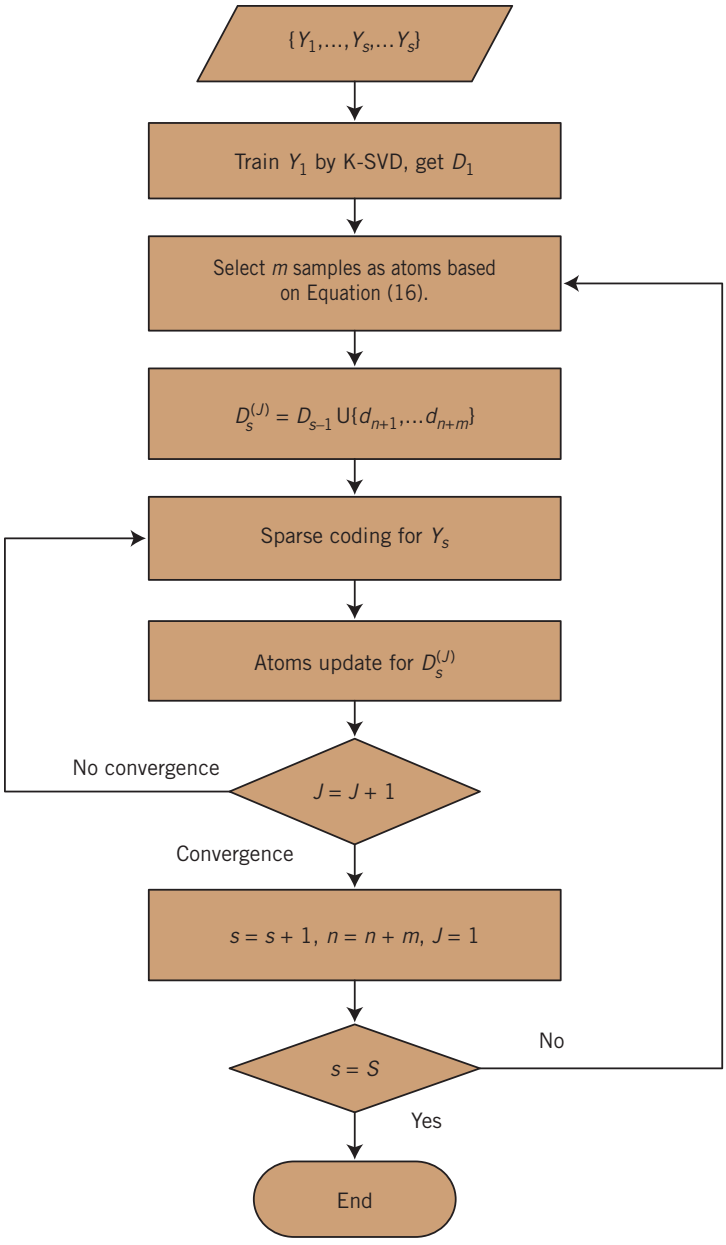

Figure 1. Flow chart of the Incremental K-SVD (IK-SVD) algorithm.

$$\min_{X_s} \left\| Y_s - D_s^{(J)} X_s \right\|_2^2 + \lambda \left\| X_s \right\|_0. \quad (20)$$

4. In the atoms-update stage, for each new atom d_k in dictionary $D_s^{(J)}$, where $k = n + 1, \ldots, n + m$, update it as follows: Define the group of examples that use this atom d_k, $\omega_k = \left\{ i \middle| 1 \le i \le r, \alpha_T^k(i) \ne 0 \right\}$. Compute the overall representation error matrix by Equation 11, and get E_s^k. Next, construct $\hat{E}_s^k$ by choosing only the columns corresponding to ω_k within E_s^k. Apply SVD decomposition $\hat{E}_s^k = U \Lambda V^T$ and choose the first column of U to be the updated atom d_k. Update the coefficient vector α_T^j to be the first column of V multiplied $\Lambda(1, 1)$.
5. Update the dictionary.
6. Set $J = J + 1$. Repeat steps 3 through 6 until convergence.
7. $s = s + 1$, $n = n + m$, and $J = 1$. Repeat steps 2 through 7 until all the data in $\{Y_1, \ldots, Y_S\}$ are trained.

Convergence means that, in the interloop, the error satisfies $\left\| Y_s - D_s^{(J)} X_s \right\|_2^2 \le \sigma$ or the sparsity satisfies $\left\| X_s \right\|_0 \le k$. Figure 1 shows the flow chart of the algorithm.

Now, we can find the differences between the proposed IK-SVD, ODL,[1] and RLSDL algorithms.[2] For IK-SVD, we always update the atoms based on the new sample data that can't be well represented by the old dictionary. The number of the atoms for IK-SVD changes in the training process, which makes IK-SVD very flexible. The atoms-updating stage of ODL is similar to the gradient descendent, therefore, the most important thing is to find an appropriate gradient that fits both new sample data and old sample data. The RLSDL algorithm is the same as the recursive least squares algorithm for adaptive filtering. Thus, a forgetting factor is very important to the atoms-update stage of RLSDL.

Experiments and Results

In our experiments, we used the image dataset of the Landsat satellite, which represents the world's longest acquired collection of moderate-resolution remote-sensing data. In the past four decades, since July 1972, the imagery datasets from Landsat 1 to Landsat 8 satellite missions have provided a unique and extremely rich resource for research on agriculture, geology, forestry, regional planning, education, mapping, and global change.

We've included different Landsat satellite datasets in our experiments because they have different resolution and spectral characteristics. Because the Landsat satellite series has continuously acquired image data for four decades, the whole image data volume is large enough to qualify as Big Data. In general, as we've mentioned earlier, it's hard to precisely model remote-sensing Big Data. Accordingly, we trained the samples in our experiments by randomly selecting Landsat data subsets.

For our purposes, we compared our proposed algorithm, IK-SVD, with the two dictionary learning algorithms that we mentioned in the last section: ODL[1] and RLSDL.[2] The volume of the global Landsat data was so large, however, that it was unrealistic to put all of them into the dictionary learning process for the three algorithms. However, because of the highly redundant nature of the massive spatiotemporal remote-sensing image set, we could compare the three methods by randomly selecting sample data from Landsat data. We compared the performance of the algorithms in two respects: one was the precision of the reconstruction; the other was the sparse extent of the decomposition. The two characteristics are mutually restrictive. Therefore, we compared one feature of these different algorithms while the other feature was fixed. For convenience of comparison, we used the OMP method to reconstruct all three methods. This meant that we trained the dictionary using different models, but solved the sparse coefficients using the same reconstruction algorithm.

A subset of the Landsat global image dataset was selected for the validations. The data subset we used in these experiments ranges from the years 2008 and 2009, which cover the whole area of China—more than 9 million square kilometers. In this data subset, only one multispectral image set was selected for every location of earth's surface. The test dataset didn't include some image data that either were destroyed or had too much cloud cover. The dataset volume with all bands was about 650 Gbytes. We trained the dictionary by randomly selecting 30-Gbyte data samples within the dataset, and we validated the performance of the three algorithms by randomly selecting another 10-Gbyte data samples for each test.

Remote-sensing Big Data from Landsat satellites contain many long temporal sequence datasets for many locations of the earth's surface. We selected the data subset for the Beijing area, in northern China, which covers 16,411 square kilometers. The time ranges from years 1983 to 2013, and again some data with too much cloud cover were removed from the dataset. The area features many forests, cities, and mountains, which, along with the high degree of climatic seasonality, makes the texture information very rich. The volume of the tested data subset is about 110 Gbytes.

We trained the dictionary by randomly selecting 3-Gbyte data samples within the dataset, and we validated the performance of the three algorithms by randomly selecting another 5-Gbyte data samples for each different test parameter. In our experiments, we set the size of the incremental data as $8 \times 8 \times 100B$ for each iteration for all three methods.

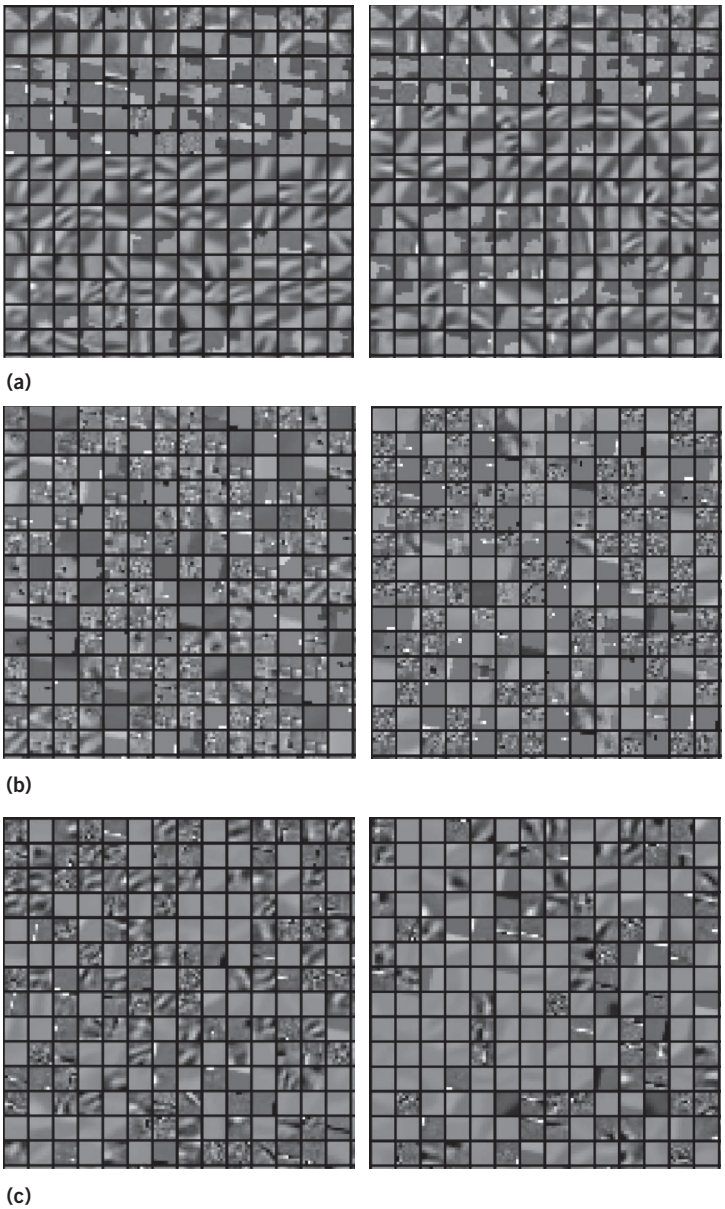

Figure 2. Different dictionaries trained by sample data. (a) IK-SVD, (b) Recursive Least Squares Dictionary Learning (RLSDL), and (c) On-Demand Localization (ODL). The atoms in the left column were trained with constraint parameter $\sigma = 10$; atoms in the right column were trained with $\sigma = 20$. Some atoms exhibit too much noise in RLSDL, and some atoms appear overly smooth with very few textures in ODL. The texture of the IK-SVD atoms is noticeably richer than the texture of those trained by the other two dictionaries.

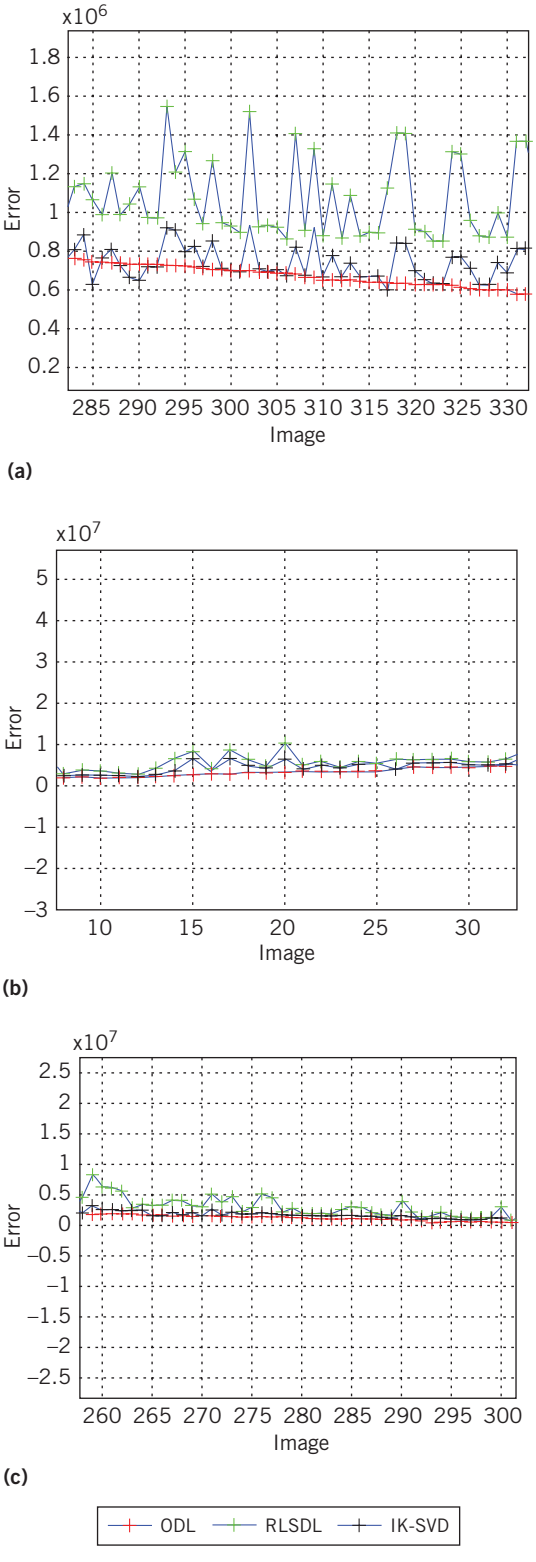

Figure 3. Precision comparison of dictionary reconstructions with decomposition sparsity controlled. We show the sparsity at (a) 5 percent, (b) 10 percent, and (c) 15 percent.

Figure 2 shows the dictionaries trained by different methods. Figure 2a shows the atoms trained by the proposed IK-SVD dictionary, Figure 2b shows the atoms trained by RLSDL, and Figure 2c shows the atoms trained by ODL. The number of atoms that were trained ranged from 150 to 3,000, but for practical reasons only a very small number of atoms is shown.

On the basis of what the atoms looked like, it was impossible to precisely judge which method performed better in training. However, we could infer that they exhibited different effectiveness and adaptiveness for different textures of large datasets. Furthermore, we observed that some atoms had too much noise in RLSDL, and in ODL some atoms were overly smooth and contained very few textures. The texture of the IK-SVD atoms, however, appeared noticeably richer than those trained by the other two dictionaries.

Unlike some predefined atoms of the analytic dictionary, the atoms in the adaptively learned unanalytic dictionary changed with the sparsity and presentation precision. We found some differences between atoms with different representation precision. In Figure 2, on the left, the error $\sigma = 10$, and on the right $\sigma = 20$. Therefore, we can observe that the features in the right column in Figure 2 are smoother than those in the left column.

In Figure 3, the precision of the reconstruction by different methods is compared. When comparing reconstruction errors in Figure 3, for an arbitrary image data subset $Y_s = \{y_s, \ldots, y_r\}$ the error is defined as

$$E = \sum_{i=1}^{r} \|y_i - D\alpha_i\|_2^2. \quad (21)$$

In these experiments, while training the atoms, we controlled the decomposition sparsity for each algorithm and compared the errors E in Equation 21 for different methods. Therefore, it's the constrained optimal problem as Equation 1 showed. To better validate the performance of the IK-SVD, ODL, and RLSDL methods while training the dictionary, we set the representation's sparsity at 5 percent, 10 percent, and 15 percent. For ODL and RLSDL, it was easy to set the sparsity for training. But for our proposed IK-SVD algorithm, because we dynamically introduced new atoms to the model, we needed to set the threshold for when new atoms should be added.

For a set of samples Y_{s+1}, we used OMP to sparsely decompose the samples to meet the controlled sparsity k. If the peak signal-to-noise ratio for the sparse coding stage was smaller than 34 decibels (dB), we introduced new atoms into the dictionary. For Big Data with an unlimited number of images, it's impossible to use all the data as samples, so we randomly selected sample data just from the two datasets as already mentioned and experimented on both long temporal sequence data and large area data.

When we controlled the coefficient sparsity in Figures 3a–3c (between 5 and 15 percent), we saw that the precision of IK-SVD fell roughly between that of ODL and RLSDL. Most IK-SVD results had higher precision than RLSDL but were close to that of ODL, although we selected different samples and set different sparsities.

Figure 4 shows how the coefficient sparsity of our three dictionary training methods compared. In these experiments, we controlled the error or Peak Signal-to-Noise Ratio (PSNR) of the sparse decomposition for each algorithm. Although the reconstruction error was fixed, the sparsity for the coefficients of every sample image differed for the three methods. To comprehensively validate the algorithms' performance, we also tested the sparsity with the reconstruction error $\sigma = 10$, $\sigma = 15$, and $\sigma = 20$. In this case, unlike the work shown in Figure 3, we experimented separately on long temporal sequence data and large-area data. Therefore, the left side of Figures 4a–4c shows the results of the Beijing area data subset for the years 1983 to 2013, and the right side of Figure 4a–4c shows the results of the entire China area data subset for whole-year data from 2008 and 2009. We can see that, for both long temporal sequence data and large-area data, the IK-SVD method's nonzero coefficients are fewer than with ODL and RLSDL. Furthermore, IK-SVD's better sparsity feature is relatively steady, but is not obviously affected by the representation precision σ.

We also compared the time consumption (time spent training) for the three methods, as Table 2 shows. It's hard to fairly compare the methods' speed because of their different program styles, data structures, and I/O scheme. However, since the original intention of ODL and RLSDL was to design an algorithm competent to deal with unlimited large datasets, it was worth testing the algorithms using a large dataset. If we trained a large number of atoms, it should obviously have taken more time than a smaller number. Table 2 shows that, based on our experiments, when the number of atoms for RLSDL exceeded 500, the algorithm's training time was unacceptable. Therefore, for Beijing-area data, we had to set the number of atoms for RLSDL at 150 to make its training time acceptable.

The representation precision was an important factor for training time consumption, in dictionary learning. Very few errors made the proposed IK-SVD method create more atoms, and it slowed the training. In addition, controlling the sparsity also affected the training speed. The more the sparsity constraints controlled, the less time was used. We also found that the RLSDL's performance wasn't steady, as the time consumption dramatically increased and became unacceptable with the increasing number of atoms. For the same error or sparsity, ODL was very fast when the number of the atoms was small, but when the atoms exceeded 2,600, ODL was slower than IK-SVD. The fixed number of atoms made ODL obviously faster with fewer atoms. However, for IK-SVD, more atoms joined the new training process, which meant that the acceleration of IK-SVD was not as obvious as with ODL when the number of atoms decreased.

To sparsely represent the spatiotemporal remote-sensing Big Data, we extended the classical K-SVD dictionary learning method. We constructed a new object function for big datasets, and introduced the data samples into the learning process group by group. In the computation, the model mainly focused on the incremental parts that are hard to sparsely decompose using the last dictionary of the last iteration. New atoms were added for current data samples, and an active learning scheme based on maximum mutual information was employed to determine the initial value of the new atoms. We tested the proposed method on two data subsets from Landsat satellites: one a long temporal sequence on a small area; the other, a large area over a two-year period. The experiments validated the proposed method's good performance on both decomposition sparsity and reconstruction precision. We found that, while controlling the error of the training process, the proposed IK-SVD always achieves sparser representation for a spatiotemporal remote-sensing big dataset.

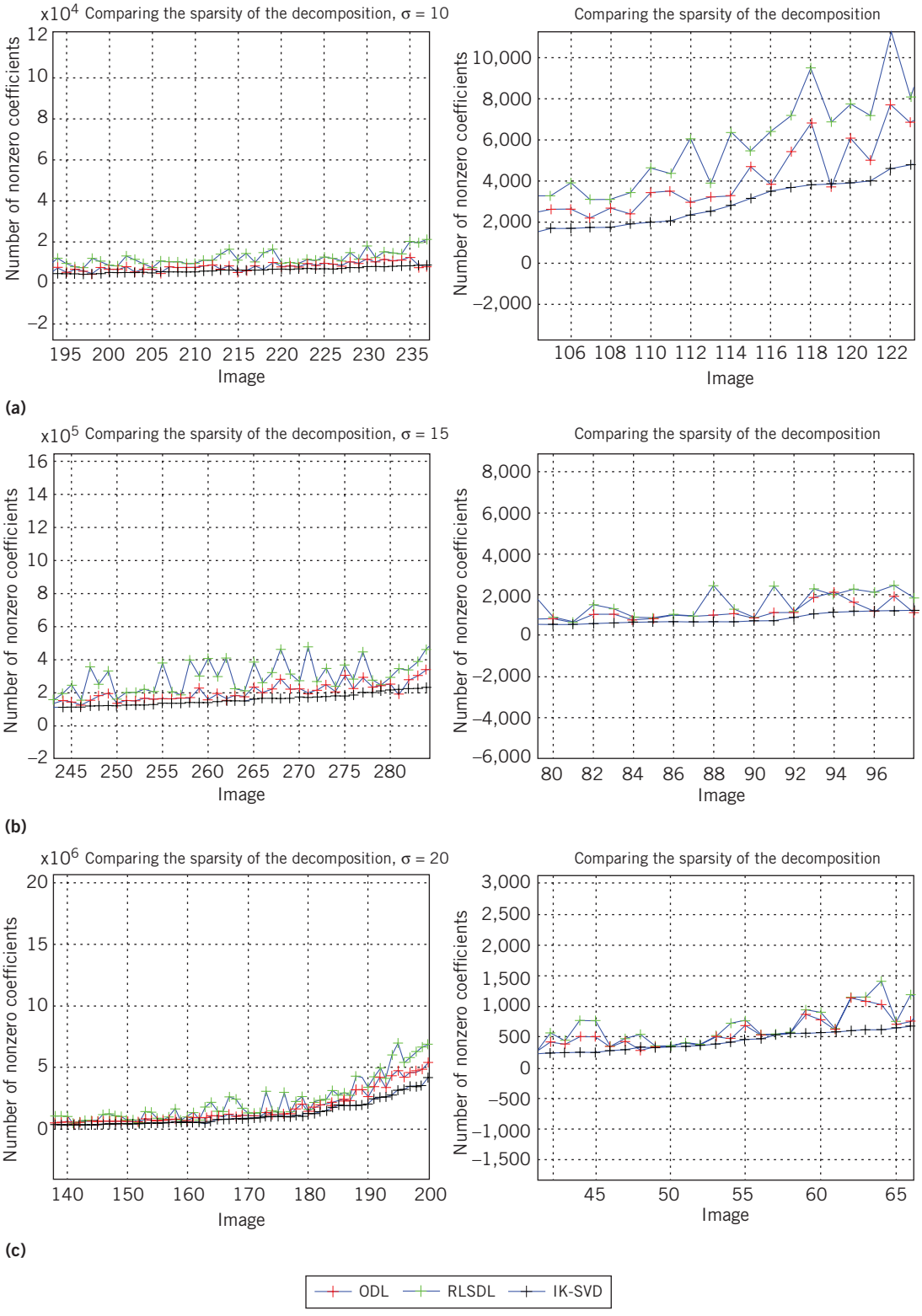

Figure 4. Sparsity of different methods while controlling the error. For each row, sparsity is (a) 10 percent, (b) 15 percent, and (c) 20 percent. The left side shows the results of the Beijing area data subset for the years 1983 to 2013, and the right side shows the results of the entire China area data subset for whole-year data from 2008 and 2009.

Table 2. The number of atoms and the training time for different methods and datasets.

Criteria			Sparsity (%)			Error (%)		
Data	Method		5	10	15	10	15	20
BJ*	IK-SVD	atoms	895	205	175	2,610	1,260	410
		time	**1.7h***	2.8h	3.5h	**10h**	4h	1.5h
	ODL	atoms	895	205	175	2,610	1,260	410
		time	2.5h	1.3h	0.7h	13.3h	2.8h	1.0h
	RLSDL	atoms	150	150	150	150	150	150
		time	3.3h	4.2h	4.5h	4.1h	1.9h	0.8h
CH*	IK-SVD	atoms	2,140	805	505	2,730	1,640	770
		time	10.2h	5.9h	5.1h	**11.4h**	5.5h	1.75h
	ODL	atoms	2,140	805	505	2,730	1,640	770
		time	**9.6h**	2.3h	1.6h	16.1h	5.7h	2.8h
	RLSDL	atoms	2,140	805	505	2,730	1,640	770
		time	—	—	45.0h	—	—	28.5h

**BJ* = dataset from Beijing area; *CH* = dataset from China area; *h* = hour.*

Furthermore, while controlling the sparsity of the training, we also found that the precision of the proposed IK-SVD algorithm generally falls between that of the ODL and RLSDL methods.

Sparse coding stages of these methods are very similar to each other. It is the relative complex atoms updating scheme that makes IK-SVD slower than ODL in the experiments on controlling sparsity. Therefore, in future work we will focus on promoting the computational efficiency of the atoms' updating in the proposed IK-SVD. Furthermore, the number of atoms also seriously influences the speed of the sparse representation. In future work, we will also consider how to merge and split the atoms in the dictionary. ■

Acknowledgments

Lajiao Chen is the corresponding author for this article.

References

1. J. Mairal et al., "Online Dictionary Learning for Sparse Coding," *ICML Proc. 26th Ann. Int'l Conf. Machine Learning*, ACM, 2009, pp. 689–696.
2. K. Skretting and K. Engan, "Recursive Least Squares Dictionary Learning Algorithm," *IEEE Trans. Signal Processing*, vol. 58, no. 4, 2010, pp. 2121–2130.
3. D.L. Donoho, "Compressed Sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, 2006, pp. 1289–1306.
4. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, 2006, pp. 4311–4322.
5. M. Zhou et al., "Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images," *IEEE Trans. Image Processing*, vol. 21, no. 1, 2012, pp. 130–144.
6. D.L. Donoho et al., "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Trans. Information Theory*, vol. 58, no. 2, 2012, pp. 1094–1121.
7. D. Needell and J.A. Tropp, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," *Comm. ACM*, vol. 53, no. 12, 2010, pp. 301–321.
8. D. Tuia et al., "Active Learning Methods for Remote Sensing Image Classification," *IEEE Trans. Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2218–2232, 2009.
9. J. Li, J. Bioucas-Dias, and A. Plaza, "Hyperspectral Image Segmentation Using a New Bayesian

Approach with Active Learning," *IEEE Trans. Geoscience and Remote Sensing*, vol. 49, no. 10, 2011, pp. 3947–3960.

Lizhe Wang is a professor at the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China, and the ChuTian Chair Professor at the School of Computer Science, China University of Geosciences. His research focuses on high-performance geocomputing and spatial information processing. Wang has a PhD in applied computer science from University Karlsruhe (now Karlsruhe Institute of Technology). Contact him at lizhe.wang@gmail.com.

Ke Lu is a professor at the College of Engineering and Information Technology, University of the Chinese Academy of Sciences, Beijing, China. His research interests include remote sensing image processing, signal processing, and medical image processing. Liu has a PhD in computer science from Northwest University. Contact him at luk@ucas.ac.cn.

Peng Liu is an associate professor at the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. His research interests are focused on remote-sensing image processing, scientific computation, and compressive sensing. Liu has a PhD in signal processing from the Chinese Academy of Sciences. Contact him at pliu@ceode.ac.cn.

Rajiv Ranjan is a research scientist and a Julius Fellow at CSIRO Computational Informatics, Canberra, Australia (formerly known as CSIRO ICT Centre). His expertise is in datacenter cloud computing, application provisioning, and performance optimization. Ranjan has a PhD in engineering from the University of Melbourne. Contact him at rajiv.ranjan@csiro.au

Lajiao Chen is an assistant professor at the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. Her research interests are focused on geocomputing, as well as geographic information systems/remote sensing techniques and their application to environmental modeling. She has a PhD in geographic information systems from the Chinese Academy of Sciences. Contact her at chenlajiao@ceode.ac.cn.